Random Balance ensembles for multiclass imbalance learning

Rodriguez, Juan; Diez-Pastor, Jose-Francisco; Arnaiz-Gonzalez, Alvar; Kuncheva, Ludmila

Knowledge-Based Systems

DOI:
10.1016/j.knosys.2019.105434

Published: 06/04/2020

Peer reviewed version

Random Balance Ensembles for Multiclass Imbalance Learning

Juan J. Rodríguez[a], José-Francisco Díez-Pastor[a], Álvar Arnaiz-González[a], Ludmila I. Kuncheva[b]

[a] *Universidad de Burgos, Escuela Politécnica Superior, Avda. Cantabria s/n, 09006 Burgos, Spain*
[b] *Bangor University, Dean Street, Bangor Gwynedd, LL57 1UT, United Kingdom*

Abstract

Random Balance strategy (RandBal) has been recently proposed for constructing classifier ensembles for imbalanced, two-class data sets. In RandBal, each base classifier is trained with a sample of the data with a random class prevalence, independent of the a priori distribution. Hence, for each sample, one of the classes will be undersampled while the other will be oversampled. RandBal can be applied on its own or can be combined with any other ensemble method. One particularly successful variant is RandBalBoost which integrates Random Balance and boosting. Encouraged by the success of RandBal, this work proposes two approaches which extend RandBal to multiclass imbalance problems. Multiclass imbalance implies that at least two classes have substantially different proportion of instances. In the first approach proposed here, termed Multiple Random Balance (MultiRandBal), we deal with all classes simultaneously. The training data for each base classifier are sampled with random class proportions. The second approach we propose decomposes the multiclass problem into two-class problems using one-vs-one or one-vs-all, and builds an ensemble of RandBal ensembles. We call the two versions of the second approach OVO-RandBal and OVA-RandBal, respectively. These two approaches were chosen because they are the most straightforward extensions of RandBal for multiple classes. Our main objective is to evaluate both approaches for multiclass imbalanced problems. To this end, an experiment was carried out with 52 multiclass data sets. The results suggest that both MultiRandBal, and OVO/OVA-RandBal are viable extensions of the original two-class RandBal. Collectively, they consistently outperform acclaimed state-of-the art methods for multiclass imbalanced problems.

Keywords: classifier ensembles, imbalanced data, multiclass classification.

1. Introduction

In classification tasks, a data set is imbalanced when the class proportions are substantially different [1, 2, 3, 4, 5], [6]. Originally, the main objective in classification was to have models with good accuracy, but in an imbalanced data set, the accuracy can be good when the instances in the minority classes are seldom predicted or even ignored. In many imbalanced problems, such as diagnosis, fault and fraud detection, it is particularly important to correctly predict the minority instances. Hence, classification methods that

were designed without taking into account the imbalance, as is the case with standard methods, may have difficulties with this type of data.

Many approaches have been proposed for dealing with imbalanced data sets, mostly focused on two-class problems, with much less attention to the multiclass case [2, 7]. Examples of multiclass imbalance problems include protein classification [8, 9], welding flaws classification [10], fault diagnosis of gearboxes [11], pediatric brain tumors [12], hyperspectral image classification [13], text categorization [14], and activity recognition [15].

Branco et al. [2] group the imbalance learning approaches into four categories: data pre-processing, special-purpose learning, prediction post-processing and hybrid. This categorization is also applicable to the multiclass case. The data pre-processing approaches usually change the training data distribution so that any standard method for constructing classifiers can be used thereafter, and the distribution change biases the classifier towards favouring the prediction of chosen classes. The special-purpose learning approaches adapt existing algorithms to deal adequately with imbalanced data. In the prediction post-processing category, a standard classifier is constructed using the original training data, and the predictions given by the classifier are subsequently modified according to the data imbalance. Hybrid methods combine approaches of the previous categories.

An alternative grouping of the imbalance learning approaches into four levels is proposed by Galar et al. [16, 17]: data level, algorithm level, cost-sensitive learning level and ensemble learning level. The first two levels correspond to the first two categories proposed in [2]. In the cost-sensitive learning group [18], errors have different costs depending on the actual and predicted classes, and the objective is to minimise the cost instead of maximise the accuracy. For imbalanced data, greater cost is assigned to errors where a minority class instance is predicted as belonging to the majority class. At the ensemble level, methods for constructing classifiers ensembles [19] are combined with approaches for imbalance learning. When constructing the ensemble, approaches from other categories can be applied, such as changing the class distributions or using cost-sensitive learning.

Random Balance [20] (RandBal) is an ensemble data-preprocessing strategy. The class proportion is chosen randomly for each classifier. Such an approach would be unsuitable for a standalone classifier, but very useful for a classifier which is a part of an ensemble. One of the requirements for constructing successful ensembles is that the member classifiers are diverse; the other is that they are accurate. Changing class distributions contributes to the diversity.

Random Balance Boost (RandBalBoost) [20] is a hybrid method which combines RandBal with AdaBoost [21]. It is a hybrid method because the data pre-processing approach of Random Balance is combined with a special-purpose modification of AdaBoost.

RandBal was originally proposed for binary tasks. Here we extend this approach to multiple classes. The multiclass task is more complex than the binary task [22, 7], starting with the choice of performance

measures. In multiclass problems, it is possible to have several minority classes, several majority classes or both. One class can be simultaneously in minority, balanced, and in majority with respect to other classes. The purpose in imbalanced binary tasks is to improve the performance with respect to the minority class without harming too much the performance with respect to the majority class. Usually the class of interest is the minority class, and high accuracy in recognising its instances is paramount. In multiclass imbalanced problems, it is not so clear to what extent one class should be preferred to another.

The two main approaches to extending a two-class classifier model to multiple classes are: (1) modify the model to accommodate more than two classes, and (2) run the model for pairs of classes and combine the decisions of the individual classifiers. The second approach is further subdivided depending on how the pairs of classes are formed: one-versus-one, one-versus-all, error-correcting output codes (ECOC) and more [23, 24]. In this study we extend RandBal using both approaches and compare the proposed variants on 52 data sets with the aim of determining whether its good results for binary problems are maintained when considering multiclass problems.

The rest of the paper is organised as follows. Section 2 reviews current approaches for classification of imbalanced multiclass data sets. Section 3 shows the extension of the Random Balance method to the multiclass case. Our experimental set-up is presented in Section 4, while the results are shown in Section 5. Finally, Section 6 offers some concluding remarks and possible future works.

2. Multiclass Imbalanced Classification

This section presents state-ot-the-art approaches for classification of imbalanced data sets with more than two classes. They are divided into groups although many approaches can be included in several groups. Fernández et al. [7] offer a detailed review on the subject. Table 1 shows some properties of these approaches. Some of these properties are marked as optional because the corresponding methods have been used in the included references both with and without the property. For instance in KSMOTE undersampling can be used in some cases.

Data level. Data-preprocessing approaches for multiclass imbalanced problems have been considered by many authors [25], [26], [28], [29] and [30].

KSMOTE was proposed by Prachuabsupakij and Soonthornphisaj [25]. It uses k-means clustering to divide the instances into two clusters, and subsequently create data subsets with various (non-random) proportions. KSMOTE was compared to Random Forest, SMOTE, one-vs-all (OVA), one-vs-one (OVO), OVA with SMOTE and OVO with SMOTE. KSMOTE and OVO with SMOTE achieved the best results in the experiment.

SCUT is a hybrid sampling method proposed by Agarwal et al. [26] for balancing the examples in multiclass data sets. The minority classes are oversampled generating synthetic examples with SMOTE while

Table 1: Methods for imbalanced multiclass classification. The properties of the methods are marked with ✓. If a method was used both with and without a property, we use (✓).

Method	References	Undersampling	Oversampling	One-vs-all	One-vs-one	Ensemble	Bagging	Boosting	Cost sensitive
KSMOTE	[25]	(✓)	✓			✓	✓		
SCUT, SMOTE and cluster-based undersampling	[26]	✓	✓						
MDO, Mahalanobis based oversampling	[27, 28, 29]		✓			(✓)		(✓)	
SMOM, synthetic oversampling for multiclass	[30]		✓			(✓)		(✓)	
Hellinger distance decision trees	[31]			(✓)					
Dynamic sampling for multilayer perceptrons	[32]	✓	✓						
Deep MLPs for imbalance	[33]								
AdaC2.M1, cost-sensitive boosting	[34]					✓		✓	✓
Cost sensitive OVO ensemble	[35]				✓	✓			✓
Cost-Sensitive neural networks with binarization	[36]				✓				✓
OVA with hybrid sampling	[9]	✓	✓	✓		✓			
OVO fuzzy rough set	[37]				✓				
Binarization with over/undersampling	[38]	(✓)	(✓)	(✓)	(✓)				
Instance weighting (cost-sensitive)	[38]			(✓)	(✓)				✓
UnderBagging	[39, 17]	✓			(✓)	✓	✓		
SMOTEBagging	[40, 17]		✓		(✓)	✓	✓		
RUSBoost	[41, 17]	✓			(✓)	✓		✓	
SMOTEBoost	[42, 17]		✓		(✓)	✓		✓	
SMOTE+AdaBoost	[17]		✓		(✓)	✓		✓	
EasyEnsemble	[43, 17, 44]	✓			(✓)	✓		✓	
Binarization with boosting and oversampling	[45]		✓	✓		✓		✓	
Diversified ECOC	[46]					✓			
RAMOBoost	[47, 32]		✓			✓		✓	
AdaBoost.NC	[48, 17, 38, 44]	(✓)	(✓)	(✓)	(✓)	✓		✓	
Probability threshold Bagging	[49]					✓	✓		
Dynamic ensemble selection	[44]	✓	✓						
Multiclass Roughly Balanced Bagging	[50, 49]	(✓)	(✓)			✓	✓		

the majority classes are undersampled using clustering. SCUT was compared to SMOTE and random undersampling, using decision trees, support vector machines, naïve Bayes and nearest neighbour as classifiers. Although there was no clear preference of one sampling method over another, SCUT was found suitable for domains where the number of classes is high and the levels of imbalance vary considerably.

Abdi and Hashemi proposed an oversampling technique inspired by the Mahalanobis distance [28], MDO. The artificially generated examples for a chosen minority class have the same Mahalanobis distance from the class mean as the other examples from this class. In this way, the covariance structure of the data in minority classes is preserved. The method compared favourably to other oversampling methods (random oversampling, SMOTE, Borderline-SMOTE [51], and ADASYN [52]), using decision trees, nearest neighbour, and rules as classifiers trained with balanced data sets (synthetic examples for each class are generated until they have as many examples as the most frequent class). An adaptive variant of MDO is proposed by Yang et al. [29]: the method is adapted to mixed-type data sets. The class distribution is partially balanced and

the method used to generate synthetic instances is optimised.

A variant of SMOTE for multiclass, SMOM, is proposed by Zhu et al. [30]. As in SMOTE, synthetic instances are obtained from real instances. An instance is selected randomly and one of its neighbours is selected randomly, but in SMOM the selection is based on weights given to the neighbours; safer neighbour directions are more likely to be selected. The weights' purpose is to avoid over generalization. The weights are based on the class distribution of the instances in the neighbourhood of the line that connects the instance with its neighbour.

Sáez et al. [22] found that oversampling benefits from distinguishing between four example types: safe examples, borderline examples, rare examples and outliers. The type of an example depends on the classes of the examples in its neighbourhood. The authors investigated the effect of oversampling of different configurations of example types. They found that the best configuration is data dependent. Configurations that were reported to be successful in general are characterised by leaving safe examples intact, e.g., processing only the rare examples or only the borderline examples.

Algorithm level. Publications reporting methods adapted to multiclass imbalance are [31], [32] and [33].

Decision trees for multiclass imbalance problems are considered in [31]. A multiclass splitting criterion is proposed, based on Hellinger distance. The results of these trees are better than for standard decision trees, but they are outperformed by OVA or ECOC of decision trees. Nevertheless, a single tree is faster and more comprehensible.

A dynamic sampling method was proposed for the multilayer perceptron (MLP) neural network [32]. The sampling is integrated within the training process. For each epoch of the training process, each example is assigned a probability of being used to update the model: examples misclassified by the current model are given probability of one, whereas the probability for correctly classified examples depends on the confidence of the model in its prediction, and on the prior probability of the class of the example. Using 20 multiclass imbalanced data sets, the method was compared to preprocessing methods (random undersampling and oversampling), a method akin to active learning (examples with the smallest difference between the two highest neurons outputs are used to update the model), three representative cost-sensitive methods, and a method based on boosting (RAMOBoost [47]). Better results were reported with the dynamic sampling on most data sets.

Another approach based on MLPs is proposed by Diáz-Vico et al. [33]. These MLPs are large, fully connected and also can be deep. They use ReLU activations, softmax outputs and categorical cross-entropy loss.

Cost-sensitive. Cost-sensitive approaches have also been proposed [34], [35], [36].

A cost-sensitive boosting algorithm for multiple classes, AdaC2.M1, is developed in [34]. As cost matrices usually are not available, a genetic algorithm is used to search the cost for each class.

Cost sensitive one-vs-one ensembles are proposed by Krawczyk [35]. The binary problems are solved with a cost sensitive neural network with a moving threshold. The outputs of the classifiers are scaled with a cost function. For each pair of classes, the costs are obtained automatically from the ROC curve.

Cost sensitive back propagation neural networks are combined with one-vs-one in the work by Zhang et al. [36]. The output of the nodes in the final layer are altered using a threshold moving method. Several aggregation strategies are used for combining the binary classifiers, including the dynamic selection of competent classifiers. In one-vs-one, a binary classifier is non-competent for the instances of classes that were not used to train the classifier.

Binarization. Approaches based on decomposition of the problem into binary problems have also been developed in the past [9], [38], as well as more recently [37], [17].

One-vs-all is combined with oversampling and undersampling in the work by Zhao et al. [9]. Different classifiers are obtained using different sets of features and combined in an ensemble with majority vote.

The use of decomposition techniques for multiclass imbalanced data sets is analysed by Fernández et al. [38]. These techniques are applied with undersampling, oversampling or cost-sensitive learning, for all classifier models: decision trees, support vector machines, and nearest neighbours. Specific methods for multiclass imbalance, not based on decomposition, such as AdaBoost.NC are also included in the analysis. The best global results were obtained with the one-vs-one decomposition when used either with oversampling or with the cost-sensitive learning.

Vluymans et al. combine the one-vs-one decomposition with classifiers based on fuzzy rough set theory [37]. An adaptive weighting scheme based on the imbalance ratio of the pair of classes is used for setting the binary classifiers. The predictions of the binary classifiers are combined with a dynamic aggregation method that takes into account the classes affinity (based on fuzzy rough approximation operators) of the testing instances.

Zhang et al.[17] analyse the use of the one-vs-one decomposition in the context of multiclass imbalanced problems. One-vs-one is deemed more adequate than one-vs-all because the latter introduces an artificial class imbalance. The ensemble methods used in the comparison were: UnderBagging [39], SMOTEBagging [40], RUSBoost [41], SMOTEBoost, SMOTE+AdaBoost, and EasyEnsemble [43]. Moreover, AdaBoost.NC [48] was included in the comparisons as an ensemble method not based on binary decompositions. Decision trees, neural networks and SVMs were used as base classifiers. Based on their experimental study, the authors recommended SMOTE+AdaBoost and EasyEnsemble with OVO.

The performance of some ensemble methods in multiclass imbalanced problems is studied in [48]. The authors propose to use AdaBoost.NC (a variant of AdaBoost based on Negative Correlation [53]) trained with oversampled data. This method is compared with AdaBoost (in three versions: without resampling, with random oversampling, and with random undersampling) and with SMOTEBoost [42], both using decision

trees as the base classifiers. The two ensemble methods were also used with the one-vs-all decomposition method. It was reported that the chosen decomposition did not provide any advantage over using the ensemble methods without decomposition.

A method termed "binarization with boosting and oversampling" is proposed by Sen et al. [45]. The binary problems are obtained with one-vs-all. Only the misclassified instances by the previous base classifiers are oversampled. The method is also used for semi-supervised classification. The base classifiers include neural networks, decision trees, nearest neighbours, support vector machines and random forest.

In ECOC [23] each class in the binary problems contains several classes of the original problem. That is, those binary classifiers discriminate between two *sets* of classes. In Diversified ECOC [46], the predictions of the binary classifiers are combined minimizing a weighted loss favouring the minority classes. It is also an ensemble method because, for each binary problem, several methods are used to train classifiers and the best method is selected. Hence, the selected binary classifiers may have been obtained with different methods.

Ensemble methods. Ensemble methods for multiclass imbalance problems have recently come to the fore [44], [50].

An alternative to rebalancing the data is to build the classifiers using the original imbalanced data and then apply thresholds to the continuous outputs. This approach is used with Bagging in the work by Collell et al. [49]. The thresholds are set equal to the prior probabilities of the respective classes, although for some performance measurements there could be better settings.

The use of dynamic ensemble selection has also been considered [44]. Only a subset of the classifiers in the ensemble is used for predicting the class of each instance. As in RandBal, the base classifiers are trained with data sets obtained with under and oversampling. These data sets are balanced, but their size is random[1]. For selecting the classifiers in the ensemble, the performances of the base classifiers for the nearest neighbours of the instance to classify is used.

Roughly Balanced Bagging [54] is a variant of Bagging for two-class imbalanced data. In the generated data sets, the number of instances of the minority class is the same as for the original training data. For the majority class, the number of instances of the majority class is obtained according to the negative binomial distribution with a probability for both classes of 0.5. Then, different samples will have different number of instances, but on average the number of instances of the majority class will be equal to that number for the minority class. Roughly Balanced Bagging has been extended to the multiclass case [50]. The number of instances of each class is obtained using the binomial distribution, with the same probability for all the classes. Then, on average the number of instances of each class will be the same, but in different samples the values will be different. With respect to the sample sizes, the authors propose two approaches. In the oversampling approach, the sample size is equal to the original training set size. In the undersampling

[1] In fact, they also use the term *random balance*, although for balanced data sets of different sizes.

approach, the sample size is the size of the minority class multiplied by the number of classes. In both approaches there will be over and undersampling, but one of them is predominant.

For ensemble methods, one scarcely used strategy is to train the base classifiers with different class proportions. We set to demonstrate in this paper that the use of this strategy, as is done in RandBal, could be advantageous for multiclass imbalanced problems.

Software. There are a few software packages specific for imbalanced classification. Imbalanced-learn[2] [55] is an open-source python library. It includes methods for undersampling, oversampling, combinations of oversampling and undersampling, as well as and ensemble learning methods. Several of the implemented methods support multiclass problems.

Multi-Imbalance[3] [56] is an open-source package, implemented in MATLAB and Octave, for multiclass imbalanced classification. It includes variants of OVO, OVA, ECOC, AdaBoost, decision trees, etc.

3. Random Balance Ensembles for Multiclass Imbalanced Problems

In the Random Balance ensemble method [20] for two-class imbalanced problems, the classifiers are trained on samples of the original training data, as it is done in other ensemble methods, such as Bagging [57]. The difference is that, in Random Balance, the proportions of the classes are assigned randomly for each classifier's training data, regardless of the priors in the original training data. In particular, given a data set with n instances, the transformed data set has also n instances, where the number of instances of one of the classes is a random integer k drawn from the interval $[2, n-2]$, and the remaining $n-k$ instances are from the other class. Let C_1 be the class requiring k instances in the sample, and $n_1 = |C_1|$ be the number of available instances of C_1. If $k < n_1$, the k instances are obtained by undersampling, otherwise, by oversampling. Among the many undersampling and oversampling methods, we choose the following ones: for undersampling, a random sample without replacement is taken. For oversampling, all the instances of the class are included and the necessary number of artificial instances is generated with SMOTE [58].

3.1. MultiRandBal (proposed extension #1)

RandBal can be extended to multiple classes by modifying the method itself. Examples of such extensions are rather frequent in machine learning, as illustrated by the multiclass extensions of the (originally two-class) boosting and support vector machines.

Algorithm 1 shows the pseudo-code for the proposed Random Balance sampling method for multiclass imbalanced problems. A weight is assigned to each class, randomly drawn from a uniform distribution over the interval $[0, 1]$. The weights are scaled to sum 1, and indicate the proportion of examples in the

[2] `https://github.com/scikit-learn-contrib/imbalanced-learn`, `http://imbalanced-learn.org/`
[3] `https://github.com/chongshengzhang/Multi_Imbalance`

transformed data set that will be sampled from the respective class. A minimum of two instances are required for each class. Occasionally, this may lead to the resulting data set having a few more instances than the original data set. Algorithm 2 shows the pseudo-code for the proposed Random Balance ensemble method for multiclass imbalanced problems (MultiRandBal). It simply builds each base classifier with a data set obtained with a sample obtained with Random Balance.

Algorithm 1: Random Balance sampling method for Multiclass problems.

Input: A training set $S = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\}$ where $\mathbf{x}_i \in \mathbf{X}$, $y_i \in \mathbf{Y} = \{\omega_1, \ldots, \omega_c\}$
Output: Data set S'
for $i \leftarrow 1, \ldots, c$ **do**
  $S_i \leftarrow \{(\mathbf{x}_j, y_j) | (\mathbf{x}_j, y_j) \in S, y_j = \omega_i\}$ // all examples from class ω_i
  $n_i \leftarrow |S_i|$ // number of examples of class ω_i
for $i \leftarrow 1, \ldots, c$ **do**
  $w_i \leftarrow \texttt{random-value}(0, 1)$ // weight of class ω_i
$w \leftarrow \sum_{i=1}^{c} w_i$
$S' \leftarrow \emptyset$ // new data set
for $i \leftarrow 1, \ldots, c$ **do**
  $n'_i \leftarrow \texttt{max}\left(\left[n \frac{w_i}{w}\right], 2\right)$ // new number of examples of ω_i, at least 2
  if $n'_i \leq n_i$ **then**
    $S' \leftarrow S' \cup \texttt{undersample}(S_i, n'_i)$
  else
    $S' \leftarrow S' \cup S_i \cup \texttt{oversample}(S_i, n'_i - n_i)$

Algorithm 2: Random Balance ensemble method for Multiclass problems (MultiRandBal).

Input: A training set $S = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\}$ where $\mathbf{x}_i \in \mathbf{X}$, $y_i \in \mathbf{Y} = \{\omega_1, \ldots, \omega_c\}$, ensemble size L, base learner.
Output: Ensemble E
for $t \leftarrow 1, \ldots, L$ **do**
  $S' \leftarrow \texttt{random-balance}(S)$
  $D_t \leftarrow \texttt{build-classifier}(S')$
$E \leftarrow \bigcup_{t=1}^{L} D_t$ // the estimate of $P(\omega_i|\mathbf{x})$ is $E_i(\mathbf{x}) = \frac{1}{L} \sum_{t=1}^{L} D_{t,i}(\mathbf{x})$

The transformed data set is used to train a base classifier. The prediction of classifier t in relation to class ω_i for an input $\mathbf{x}$, denoted $D_{t,i}(\mathbf{x})$ $(t = 1, \ldots, L,\ i = 1, \ldots, c)$ can be in the form of an estimate of the posterior probability $P(\omega_i|\mathbf{x})$ or a binary index containing 1 if ω_i is the predicted class, and 0, otherwise. The predictions could be combined using any method [19]; currently the average method is used, as shown in Algorithm 2.

Note that the only parameter of MultiRandBal is the ensemble size. It is expected that greater values will give better results, but the improvement decreases quickly with increasing the ensemble size. On the other hand, in order to tune the performance for a specific data set, some parameters could be introduced such as a maximum imbalance ratio allowed for each randomly sampled distribution. Restrictions related to

the true prior probabilities or chosen misclassification costs can also be imposed on the values $w_1, \ldots, w_c$.

The Random Balance ensemble method can be combined with any ensemble method: its base learner can be chosen to suit the particular ensemble or can be an ensemble method itself. For example, the combination of Bagging with Random Balance is included in the experiment later on: the ensemble size for Bagging is 100 and its base classifier is Random Balance with ensemble size 1. Then the 100 base classifiers are trained on bootstrap samples from data with different class probabilities.

Random Balance ensembles can also be combined with boosting methods. MultiRandBalBoost is based on AdaBoost.M2 (as SMOTEBoost and RUSBoost). In each boosting iteration, a data set is obtained using the Random Balance sampling method and the obtained sampled is used to build the classifier.

3.2. OVO-RandBal and OVA-RandBal (proposed extension #2)

RandBal can be straightforwardly extended to multiclass problems by using decomposition techniques, such as one-vs-one (OVO) and one-vs-all (OVA).

In the OVO decomposition, all pairs of classes are formed and a classifier is built for each pair. Thus, if there are c classes in the data, the ensemble consists of $c(c-1)/2$ classifiers. Each classifier votes for one of the classes it has been trained on. In the classical version of OVO, the resultant label is obtained by the majority vote. In Weka, ensemble probabilities are calculated from the votes. The standard two-class RandBal *sampling heuristic* of random class proportions is applied for creating the data for each classifier.

OVA creates c binary classifiers where each classifier is paired with all the remaining classes. Again, the two-class RandBal sampling is applied to the designated class and the *set* of the remaining $c-1$ classes regarded as one compound class. In doing so, some small classes may be completely wiped out in some of the training data.

For example, suppose that there are three classes, c_1, c_2, and c_3, with proportions 0.75, 0.20 and 0.05, respectively. Consider the binary classifier distinguishing between c_2 and $\{c_1, c_3\}$. Suppose that we generated random proportions whereby class c_2 is sampled with proportion 0.9, and class $\{c_1, c_3\}$ with proportion 0.1. The probability that class c_3 will not be chosen in 1 draw is $1 - 0.1 \times 0.05 = 0.9950$. If we sample 100 objects independently and with replacement, the chance that class c_3 will be completely missing from the sample is quite high, $(1 - 0.1 \times 0.05)^{100} = 0.6058$. This effect is undesirable because the vote of this classifier in favour of $\{c_1, c_3\}$ will count towards both classes but, in reality, one of the classes would not have contributed to the training. This situation with OVA is possible with other sampling methods too, but OVA-RandBal is particularly vulnerable to it due to the random proportions.

4. Experimental Set-up

This section presents the experiments and their results. The purpose of the experiment is to evaluate the performance of the two extensions of RandBal for multiclass imbalaced problems.

The distinctive feature of RandBal is that the base classifiers are trained with different class proportions. The expected effect is that the base classifiers will be more diverse, but on the other hand these arbitrarily induced (im)balance will likely harm their individual performance. The main question is whether the reduced performance of the base classifiers will translate into a superior ensemble performance owing to the much richer diversity.

First, the data sets are introduced in Section 4.1. The performance measures are described in Section 4.2. The methods and their settings are listed in Section 4.3.

4.1. Data sets

Table 2 summarises the characteristics of the data sets. These data sets were sourced from three repositories:

- KEEL data set repository [59]; we chose the data sets in the category "multiple class imbalanced problems".
- The data sets used in [22]. We refer to this repository[4] as PWR after the host university (Wrocław University of Science and Technology, Poland).
- The data sets used in [60]; we chose the multiclass data sets with an imbalance ratio of at least 2.0. We refer to this repository[5] as USC after the host university (University of Santiago de Compostela, Spain).

Many of the data sets in the three repositories are versions of data sets originally stored in the UCI Machine Learning Repository [61].

4.2. Measures

Following the literature on imbalance learning, we adopt the following classifier performance measures adapted to accommodate multiclass problems.

The Precision and Recall for class ω_i are defined as:

$$\text{Precision}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i} \qquad \text{Recall}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i} \ ,$$

where TP_i is the number of true positives (examples of class ω_i which are classified correctly), FP_i is the number of false positives (examples that are wrongly assigned to class ω_i) and FN_i is the number of false negatives (examples of class ω_i assigned to another class).

Overall accuracy is included in our experiment, as it is the most used measure in multiclass classification, although we note that it should be used with caution. Very high values of this measure can be deceiving

[4]The repository is available at `http://www.kssk.pwr.edu.pl/krawczyk/multi-over`
[5]The repository is available at `http://persoal.citius.usc.es/manuel.fernandez.delgado/papers/jmlr/`

Table 2: Characteristics of the data sets (#E: examples, #N: numeric features, #D: discrete features, #C: classes, IR: imbalance ratio). IR is defined as the number of examples of the greatest class divided by the number of examples of the smallest class.

Data set	Source	#E	#N	#D	#C	IR	Examples of each class (descending order)
annealing	USC	898	31	0	5	85.500	684 99 67 40 8
arrhythmia	USC	452	262	0	13	122.500	245 50 44 25 22 15 15 13 9 5 4 3 2
audiology-std	USC	196	59	0	18	23.500	47 45 21 20 18 8 8 4 4 4 3 2 2 2 2 2 2 2
autos	KEEL	159	15	10	6	16.000	48 46 29 20 13 3
balance	KEEL	625	4	0	3	5.878	288 288 49
cardiotocography-10clases	USC	2126	21	0	10	10.925	579 384 332 252 197 107 81 72 69 53
cardiotocography-3clases	USC	2126	21	0	3	9.403	1655 295 176
car	PWR	1728	0	6	4	18.615	1210 384 69 65
chess-krvk	USC	28056	6	0	18	168.630	4553 4194 3597 2854 2796 2166 1985 1712 1433 683 592 471 390 246 198 81 78 27
cleveland	PWR	297	13	0	5	12.308	160 54 35 35 13
contraceptive	KEEL	1473	6	3	3	1.889	629 511 333
dermatology	KEEL	366	34	0	6	5.600	112 72 61 52 49 20
ecoli	KEEL	336	7	0	8	71.500	143 77 52 35 20 5 2 2
energy-y1	USC	768	8	0	3	2.628	360 271 137
energy-y2	USC	768	8	0	3	2.026	383 196 189
flags	USC	194	28	0	8	15.000	60 40 36 27 15 8 4 4
flare	PWR	1066	0	11	6	7.698	331 239 211 147 95 43
glass	KEEL	214	9	0	6	8.444	76 70 29 17 13 9
hayes-roth	KEEL	132	4	0	3	1.700	51 51 30
heart-cleveland	USC	303	13	0	5	12.615	164 55 36 35 13
heart-switzerland	USC	123	12	0	5	9.600	48 32 30 8 5
heart-va	USC	200	12	0	5	5.600	56 51 42 41 10
led7digit	PWR	500	7	0	10	1.541	57 57 53 52 52 51 49 47 45 37
lenses	USC	24	4	0	3	3.750	15 5 4
low-res-spect	USC	531	100	0	9	138.000	276 103 90 39 7 6 6 2 2
lymphography	KEEL	148	3	15	4	40.500	81 61 4 2
molec-biol-splice	USC	3190	60	0	3	2.158	1655 768 767
new-thyroid	KEEL	215	5	0	3	5.000	150 35 30
nursery	USC	12960	8	0	5	2160.000	4320 4266 4044 328 2
oocytes-merluccius-states-2f	USC	1022	25	0	3	11.508	702 259 61
oocytes-trisopterus-states-5b	USC	912	32	0	3	37.500	525 373 14
pageblocks	KEEL	548	10	0	5	164.000	492 33 12 8 3
penbased	KEEL	1100	16	0	10	1.095	115 115 114 114 114 106 106 106 105 105
pittsburg-bridges-MATERIAL	USC	106	7	0	3	7.182	79 16 11
pittsburg-bridges-REL-L	USC	103	7	0	3	3.533	53 35 15
pittsburg-bridges-SPAN	USC	92	7	0	3	2.182	48 22 22
pittsburg-bridges-TYPE	USC	105	7	0	6	4.400	44 16 13 11 11 10
post-operative	PWR	87	0	8	3	62.000	62 24 1
primary-tumor	USC	330	17	0	15	14.000	84 39 29 28 24 24 20 16 14 14 10 9 7 6 6
shuttle	KEEL	2175	9	0	5	853.000	1706 338 123 6 2
soybean	USC	683	35	0	18	11.500	92 91 91 88 44 44 36 20 20 20 20 20 20 20 20 15 14 8
statlog-landsat	USC	6435	36	0	6	2.449	1533 1508 1358 707 703 626
statlog-shuttle	USC	58000	9	0	7	4558.600	45586 8903 3267 171 50 13 10
steel-plates	USC	1941	27	0	7	12.236	673 402 391 190 158 72 55
thyroid	KEEL	720	21	0	3	39.176	666 37 17
vehicle	PWR	846	18	0	4	1.095	218 217 212 199
vertebral-column-3clases	USC	310	6	0	3	2.500	150 100 60
wall-following	USC	5456	24	0	4	6.723	2205 2097 826 328
wine	KEEL	178	13	0	3	1.479	71 59 48
winequality-red	PWR	1599	11	0	6	68.100	681 638 199 53 18 10
yeast	KEEL	1484	8	0	10	92.600	463 429 244 163 51 44 35 30 20 5
zoo	PWR	101	0	16	7	10.250	41 20 13 10 8 5 4

because such values may result from always predicting the majority classes and ignoring minority classes of interest. The accuracy is calculated as

$$\text{Accuracy} = \frac{1}{n}\sum_{i=1}^{c} \text{TP}_i$$

where c is the number of classes and n the number of examples.

Being less sensitive to the class distributions than accuracy, Kappa (κ) has been used for multiclass classification [24].

$$\text{Kappa} = \frac{n\sum_{i=1}^{c} \text{TP}_i - \text{ABC}}{n^2 - \text{ABC}}$$

where ABC (agreement by chance) is $\sum_{i=1}^{c} (\text{TP}_i + \text{FP}_i)\,(\text{TP}_i + \text{FN}_i)$.

G-mean [34, 48] is the geometric mean of the recall values of all the classes.

$$\text{G-mean} = \left(\prod_{i=1}^{c} \text{Recall}_i\right)^{1/c}$$

The average accuracy [17] is the arithmetic mean of the recall values of all the classes.

$$\text{average-Accuracy} = \frac{1}{c}\sum_{i=1}^{c} \text{Recall}_i$$

The F-measure [11] for a class is the harmonic mean of its Precision and Recall. For multiclass data sets the arithmetic mean of the F-measure values of all the classes is used.

$$\text{F-measure} = \frac{1}{c}\sum_{i=1}^{c} \frac{2 \cdot \text{Recall}_i \cdot \text{Precision}_i}{\text{Recall}_i + \text{Precision}_i}$$

MAUC [62, 48, 28] is the average AUC (Area Under the ROC Curve) of all pairs of classes.

$$\text{MAUC} = \frac{1}{c(c-1)} \sum_{\substack{i,j=1 \\ i \neq j}}^{c} \text{AUC}(i,j)$$

where $\text{AUC}(i,j)$ is the area under the curve for the pair of classes i and j.

4.3. Methods and Settings

The experiments were performed using Weka [63]. The settings for the considered methods were the defaults in Weka, unless otherwise specified

The results were obtained with a 25×2-fold stratified cross validation. Using two-fold cross validation ensures that there will be at least one instance of each class in each fold, provided that there is more than one instance in the original data set. Weka's implementation of SMOTE uses 5 neighbours by default and for nominal attributes it uses *Value Distance Metric* (VDM).

Average ranks [64, 65] were used to compare the methods across the different data sets. For each data set, the methods are sorted from best to worst. The best method receives a rank of 1, the second best

Table 3: Ensemble methods included in the experimental study.

	Multiclass strategy		
Ensemble method	Direct	One-vs-all	One-vs-one
Existing			
(1) SMOTEBagging	(i) SMOTEBagging	(ii) OVA-SMOTEBagging	(iii) OVO-SMOTEBagging
(2) Roughly Balanced Bagging	(iv) OverMultiRoughBalBag (v) UnderMultiRoughBalBag	(vi) OVA-RoughBalBag	(vii) OVO-RoughBalBag
(3) EasyEnsemble	-	(viii) OVA-EasyEnsemble	(ix) OVO-EasyEnsemble
(4) SMOTE+AdaBoost	-	(x) OVA-SMOTE+AdaBoost	(xi) OVO-SMOTE+AdaBoost
(5) RUSBoost	-	(xii) OVA-RUSBoost	(xiii) OVO-RUSBoost
(6) AdaBoost.NC	(xiv) AdaBoost.NC	(xv) OVA-AdaBoost.NC	(xvi) OVO-AdaBoost.NC
Proposed			
(7) Random Balance	(xvii) MultiRandBal	(vxiii) OVA-RandBal	(xix) OVO-RandBal
(8) Bagging Random Balance	(xx) BagMultiRandBal	(xxi) OVA-BagRandBal	(xxii) OVO-BagRandBal
(9) Random Balance Boost	(xxiii) MultiRandBalBoost	(xxiv) OVA-RandBalBoost	(xxv) OVO-RandBalBoost

method receives a rank of 2, and so on. If there are ties, average values are assigned (e.g., if four methods achieve the top spot, each method will be assigned a rank of 2.5). The average ranks were calculated across the data sets. Adjusted p-values from Hochberg procedure [66, 65] were used to determine the significance of the rank differences.

The methods were also compared using the Bayesian Signed-Rank Test [67], the Bayesian equivalent of the Wilcoxon signed-rank test. For this test, the value of the *region of practical equivalence* (rope) was set to 0.01 for all the performance measures. Two classifiers are considered equivalent if the difference in their performance is smaller than this "rope". The test gives three probabilities: 1) one method is better than the other, 2) vice versa, or 3) they are in the "rope".

Table 3 shows the methods included in the comparison. The proposed variants are listed in the bottom part of the table. The alternative methods included in the comparison are all ensemble methods, because RandBal is an ensemble approach in itself.

The one-vs-one implementation in Weka uses the binary outputs from the member classifiers, $D_{t,i} \in \{0,1\}$, $\sum_i D_{t,i} = 1$, $t = 1, \ldots, L$, to calculate the ensemble probabilities $E_i(\mathbf{x})$. The probability that $\mathbf{x}$ comes from class ω_i is estimated as the proportion of votes for ω_i among the L member classifiers. This option was changed, using the probability outputs $D_{t,i} \in [0,1]$, $\sum_i D_{t,i} = 1$, $t = 1, \ldots, L$. This change affects mainly the MAUC measure.

The two decomposition techniques were combined with the considered ensemble methods. Ensemble size was fixed at $L = 100$. Decision trees (J48, based on C4.5 [68]) were chosen as the base classifier. They were used without pruning because this option usually gives better results with ensembles as they are more

unstable than pruned trees [19]. Moreover, in imbalanced data sets, pruning can make the prediction of minority classes less likely.

Nine methods specifically designed or adapted for multiclass imbalanced problems were chosen for our study, as explained below.

(1) SMOTEBagging [40] was included. It can handle multiclass imbalance by design. Thus, decomposition techniques may not be needed. To examine the extent of improvement offered by such techniques, we included in our experiment SMOTEBagging with and without them. This gives rise to three competing methods: (i) SMOTEBagging, (ii) OVA-SMOTEBagging, and (iii) OVO-SMOTEBagging.

(2) Roughly Balanced Bagging was included using the decomposition techniques and the two extensions for the multiclass case proposed in [50]. In the undersampling approach, the expected number of instances of each class is the minority class size. In [50] the data sets with less than 5 instances in some class were modified removing those classes. In this experiment, instead of modifying the data sets, a minimum size of 5 was enforced. Then, 4 competing methods were obtained: (iv) OverMultiRoughBalBag, (v) UnderMultiRoughBalBag, (vi) OVA-RoughBalBag and (vii) OVO-RoughBalBag.

(3) EasyEnsemble [43] and (4) SMOTE+AdaBoost were included as ensemble methods because they achieved best results when combined with one-vs-one (OVO), in [17]. AdaBoost was used with resampling [69] instead of reweighting, as this choice, typically, gives better results. The base classifier is not trained directly with the weighted instances, but with a sample from the instances. For EasyEnsemble, 10 data sets were constructed by undersampling the majority class, and for each data set, AdaBoost was trained with 10 base classifiers, hence the final ensemble also contained 100 classifiers. These methods are applied to multiclass data through the decomposition techniques, contributing 4 competing methods in our experiment: (viii) OVA-EasyEnsemble, (ix) OVO-EasyEnsemble, (x) OVA-SMOTE+AdaBoost, and (xi) OVO-SMOTE+AdaBoost.

(5) RUSBoost [41] was also included with the two decomposition techniques, giving rise to two competing methods (xii) OVA-RUSBoost and (xiii) OVO-RUSBoost.

(6) AdaBoost.NC [48] presented in Section 2, was trained with an oversampled data set, as recommended by its authors. A fully balanced data set was created by padding the smallest classes with artificial examples. As AdaBoost.NC handles imbalanced multiclass data sets by design, it was included with and without the decomposition techniques. This gives rise to three competing methods: (xiv) AdaBoost.NC, (xv) OVA-AdaBoost.NC, and (xvi) OVO-AdaBoost.NC.

(7) Random Balance was used as an ensemble method by itself (giving rise to three competing methods: (xvii) MultiRandBal, (xviii) OVA-RandBal, and (xix) OVO-RandBal), but also in combination with (8) Bagging (hence, (xx) BagMultiRandBal, (xxi) OVA-BagRandBal, and (xxii) OVO-BagRandBal).

Finally, (9) RandBalBoost combines the Random Balance strategy with Boosting, in a similar way as is done in SMOTEBoost [42] and RUSBoost [41]. In each iteration of Boosting, the pre-processing technique

(e.g., Random Balance, SMOTE, random undersampling) is applied and the obtained data set is used to train the base classifier. This combination can be used with or without the two decomposition techniques, which completes the set of 25 competing methods with (xxiii) MultiRandBalBoost (xxiv) OVA-RandBalBoost and (xxv) OVO-RandBalBoost.

The 25 competing methods can be divided into two groups: ones which use Random Balance, and ones which do not. Our hypothesis is that the methods which use Random Balance will fare better than the other group.

5. Results

Table 4 shows the ranks of the 25 methods for the six chosen performance measures [6]. The ranks are averaged across the 52 data sets, and, for each measure, the methods are sorted by average rank, from best to worst. The rows with methods which use Random Balance are shaded in all the tables.

OVA-RandBalBoost has the top rank for Accuracy and Kappa; MultiRandBal is the best for G-mean and average-Accuracy; MultiRandBalBoost is the best for F-measure; and BagMultiRandBal is the best for MAUC.

The ensemble methods with top ranks and without Random Balance turned out to be OVA-SMOTE+ AdaBoost for Accuracy, OVA-SMOTEBagging for Kappa and F-measure, OverMultiRoughBalBag for G-mean and average-Accuracy, and OVA-RUSBoost for MAUC. In general, for a given ensemble method, its combination with OVA and OVO have similar average ranks.

In several data sets, as shown in the supplementary material tables, the value for G-mean is zero for all the methods. The cause is that some particularly small classes are never correctly predicted. Some of these data sets have classes with only two instances, so one instance is always included in the training set (through stratified sampling) while the other is left in the testing set.

Table 4 also shows the average ranks[7] obtained when using the six performance measures *together*. That is, the rank for the method is averaged across the six measures. The top six ranks are for methods with Random Balance, the first three are OVA-RandBalBoost, MultiRandBalBoost and BagMultiRandBal. Among the methods which do not apply Random Balance, the top two are OVA-SMOTEBagging and OVA-RUSBoost.

Table 5 also shows the average ranks, but only for the Bagging-based ensemble methods. Table 6 shows the average ranks for the Boosting-based ensemble methods. For both groups of methods and measures the method with top rank is a method with Random Balance, with the only exception of G-mean. In these measures the differences among the ranks methods are smaller than for other measures.

As a visual summary of the average ranks, Figure 1 shows a stacked bar chart of the ensemble methods' ranks according to the six measures, with and without Random Balance. The bars in the left subplot

[6] The full set of results is available in the supplementary material.

[7] The adjusted p-values are not included because the results of the different measures are not independent.

Table 4: Average ranks. Rows for methods with Random Balance are highlighted with blue tex and grey background.

Accuracy

Method	Rank	$p_{Hochberg}$
OVA-RandBalBoost	4.2115	
OVA-BagRandBal	6.3077	0.146430
MultiRandBalBoost	6.4904	0.146430
OVO-RandBalBoost	6.7212	0.146430
OVA-SMOTE+AdaBoost	7.7500	0.056902
OVA-SMOTEBagging	8.2019	0.028495
OVA-RandBal	9.1635	0.003611
OVA-RUSBoost	9.3365	0.002689
OVO-SMOTE+AdaBoost	9.4135	0.002507
BagMultiRandBal	10.3846	0.000171
OVO-BagRandBal	10.8077	0.000049
OVO-SMOTEBagging	12.0865	0.000001
OVO-RandBal	12.3077	0.000000
SMOTEBagging	13.6923	0.000000
OVA-EasyEnsemble	13.6923	0.000000
OVO-RUSBoost	14.2115	0.000000
MultiRandBal	14.9231	0.000000
OverMultiRoughBalBag	15.9904	0.000000
OVA-RoughBalBag	16.4808	0.000000
OVO-EasyEnsemble	18.7308	0.000000
UnderMultiRoughBalBag	19.2308	0.000000
OVA-AdaBoost.NC	20.0192	0.000000
OVO-RoughBalBag	20.8173	0.000000
AdaBoost.NC	21.7019	0.000000
OVO-AdaBoost.NC	22.3269	0.000000

Kappa

Method	Rank	$p_{Hochberg}$
OVA-RandBalBoost	5.4808	
MultiRandBalBoost	6.9615	0.304935
OVA-BagRandBal	7.1538	0.304935
OVO-RandBalBoost	7.7885	0.304935
OVA-SMOTEBagging	8.4423	0.160751
OVA-RandBal	9.1346	0.056795
OVA-SMOTE+AdaBoost	9.2692	0.052032
OVA-RUSBoost	9.5000	0.037515
OVO-SMOTE+AdaBoost	9.6538	0.030703
BagMultiRandBal	9.7308	0.029114
OVO-BagRandBal	11.9038	0.000086
OVO-RandBal	12.2115	0.000034
OVA-EasyEnsemble	12.5000	0.000014
OVO-SMOTEBagging	12.5962	0.000011
SMOTEBagging	13.6923	0.000000
MultiRandBal	14.0769	0.000000
OVO-RUSBoost	14.1538	0.000000
OverMultiRoughBalBag	14.9615	0.000000
OVA-RoughBalBag	15.3462	0.000000
OVO-EasyEnsemble	17.8269	0.000000
UnderMultiRoughBalBag	18.0577	0.000000
OVO-RoughBalBag	19.8846	0.000000
OVA-AdaBoost.NC	20.9808	0.000000
AdaBoost.NC	21.3654	0.000000
OVO-AdaBoost.NC	22.3269	0.000000

All measures

Method	Rank
OVA-RandBalBoost	7.9952
MultiRandBalBoost	8.1314
BagMultiRandBal	9.1282
OVA-BagRandBal	9.5769
OVA-RandBal	9.9808
OVO-RandBalBoost	10.1314
OVA-SMOTEBagging	10.2179
OVA-RUSBoost	10.3830
MultiRandBal	11.2115
OVA-SMOTE+AdaBoost	11.6330
OVO-RandBal	12.0913
OverMultiRoughBalBag	12.4824
OVO-SMOTE+AdaBoost	12.5641
OVA-EasyEnsemble	12.7276
SMOTEBagging	12.8189
OVO-SMOTEBagging	13.1538
OVA-RoughBalBag	13.6699
OVO-BagRandBal	14.0481
UnderMultiRoughBalBag	14.4311
OVO-RUSBoost	15.3462
OVO-EasyEnsemble	15.6154
OVO-RoughBalBag	17.3029
AdaBoost.NC	18.9840
OVA-AdaBoost.NC	20.1811
OVO-AdaBoost.NC	21.1939

G-mean

Method	Rank	$p_{Hochberg}$
MultiRandBal	9.0769	
OverMultiRoughBalBag	9.9615	0.539956
UnderMultiRoughBalBag	10.6827	0.531838
OVO-EasyEnsemble	11.0385	0.522446
OVA-RUSBoost	11.0577	0.522446
OVA-EasyEnsemble	11.1154	0.522446
MultiRandBalBoost	11.2596	0.522446
BagMultiRandBal	11.3654	0.522446
OVO-RandBal	11.3942	0.522446
OVA-RoughBalBag	11.5000	0.522446
OVA-RandBal	11.5288	0.522446
OVA-SMOTEBagging	11.9519	0.510253
OVA-RandBalBoost	11.9519	0.510253
SMOTEBagging	12.0673	0.497694
OVO-RoughBalBag	12.5769	0.214394
OVO-SMOTE+AdaBoost	13.3365	0.047489
OVO-RandBalBoost	13.6635	0.023756
OVO-SMOTEBagging	13.8173	0.017384
OVA-SMOTE+AdaBoost	14.2981	0.005358
AdaBoost.NC	14.3750	0.004597
OVA-BagRandBal	14.4615	0.003821
OVO-RUSBoost	16.4423	0.000007
OVO-BagRandBal	17.4423	0.000000
OVO-AdaBoost.NC	18.1442	0.000000
OVA-AdaBoost.NC	20.4904	0.000000

average-Accuracy

Method	Rank	$p_{Hochberg}$
MultiRandBal	8.6154	
BagMultiRandBal	8.7885	0.904553
OverMultiRoughBalBag	9.8077	0.817546
MultiRandBalBoost	10.2692	0.755611
OVO-RandBal	10.7885	0.528729
OVA-SMOTEBagging	10.8462	0.528729
OVA-RandBal	10.8462	0.528729
UnderMultiRoughBalBag	11.2308	0.489912
OVA-EasyEnsemble	11.3269	0.482383
OVA-RandBalBoost	11.5000	0.410943
OVA-RUSBoost	11.7500	0.298764
OVA-RoughBalBag	11.8269	0.286880
OVO-EasyEnsemble	12.1538	0.170706
SMOTEBagging	12.2500	0.153371
OVO-RandBalBoost	12.6346	0.075030
OVA-BagRandBal	12.7692	0.060056
OVO-SMOTEBagging	13.6346	0.008100
OVO-SMOTE+AdaBoost	13.6538	0.008100
OVA-SMOTE+AdaBoost	14.7500	0.000384
OVO-RoughBalBag	15.3077	0.000067
OVO-RUSBoost	15.5385	0.000032
OVO-BagRandBal	16.5577	0.000001
AdaBoost.NC	16.6154	0.000001
OVO-AdaBoost.NC	20.4038	0.000000
OVA-AdaBoost.NC	21.1346	0.000000

F-measure

Method	Rank	$p_{Hochberg}$
MultiRandBalBoost	6.8077	
OVA-RandBalBoost	7.3077	0.729034
BagMultiRandBal	8.9231	0.285525
OVA-SMOTEBagging	8.9808	0.285525
OVO-RandBalBoost	9.1154	0.285525
OVA-RandBal	9.6923	0.228301
OVA-RUSBoost	9.7692	0.228301
OVA-SMOTE+AdaBoost	9.9808	0.195458
OVA-BagRandBal	10.1731	0.157771
OVO-SMOTE+AdaBoost	10.3846	0.118854
OVO-RandBal	11.6731	0.007494
MultiRandBal	11.8269	0.005569
OVO-SMOTEBagging	12.0192	0.003665
SMOTEBagging	12.5000	0.001043
OverMultiRoughBalBag	12.7500	0.000537
OVA-EasyEnsemble	13.9231	0.000012
OVO-BagRandBal	14.3269	0.000003
OVA-RoughBalBag	15.0769	0.000000
OVO-RUSBoost	15.0962	0.000000
UnderMultiRoughBalBag	16.2115	0.000000
OVO-EasyEnsemble	16.9615	0.000000
OVO-RoughBalBag	18.5385	0.000000
AdaBoost.NC	19.9231	0.000000
OVA-AdaBoost.NC	21.1731	0.000000
OVO-AdaBoost.NC	21.8654	0.000000

MAUC

Method	Rank	$p_{Hochberg}$
BagMultiRandBal	5.5769	
OVA-BagRandBal	6.5962	0.480099
MultiRandBalBoost	7.0000	0.480099
OVA-RandBalBoost	7.5192	0.480099
MultiRandBal	8.7500	0.111690
OVA-RandBal	9.5192	0.031541
OVO-RandBalBoost	10.8654	0.001490
OVA-RUSBoost	10.8846	0.001490
UnderMultiRoughBalBag	11.1731	0.000846
OverMultiRoughBalBag	11.4231	0.000460
OVA-RoughBalBag	11.7885	0.000168
SMOTEBagging	12.7115	0.000008
OVA-SMOTEBagging	12.8846	0.000005
OVO-BagRandBal	13.2500	0.000001
OVA-SMOTE+AdaBoost	13.7500	0.000000
OVA-EasyEnsemble	13.8077	0.000000
OVO-RandBal	14.1731	0.000000
OVO-SMOTEBagging	14.7692	0.000000
OVO-RUSBoost	16.6346	0.000000
OVO-RoughBalBag	16.6923	0.000000
OVO-EasyEnsemble	16.9808	0.000000
OVA-AdaBoost.NC	17.2885	0.000000
OVO-SMOTE+AdaBoost	18.9423	0.000000
AdaBoost.NC	19.9231	0.000000
OVO-AdaBoost.NC	22.0962	0.000000

Table 5: Average ranks for the Bagging-based ensemble methods. Rows for methods with Random Balance are highlighted.

Accuracy

Method	Rank	$p_{Hochberg}$
OVA-BagRandBal	2.1442	
OVA-SMOTEBagging	3.0192	0.140581
BagMultiRandBal	4.0673	0.002401
OVO-BagRandBal	4.5000	0.000218
OVO-SMOTEBagging	4.8365	0.000023
SMOTEBagging	5.4038	0.000000
OverMultiRoughBalBag	6.8173	0.000000
OVA-RoughBalBag	7.0000	0.000000
UnderMultiRoughBalBag	8.2596	0.000000
OVO-RoughBalBag	8.9519	0.000000

Kappa

Method	Rank	$p_{Hochberg}$
OVA-BagRandBal	2.5192	
OVA-SMOTEBagging	3.2308	0.230785
BagMultiRandBal	3.8654	0.046765
OVO-BagRandBal	4.9423	0.000135
OVO-SMOTEBagging	5.1538	0.000036
SMOTEBagging	5.5577	0.000002
OverMultiRoughBalBag	6.4808	0.000000
OVA-RoughBalBag	6.6731	0.000000
UnderMultiRoughBalBag	7.8846	0.000000
OVO-RoughBalBag	8.6923	0.000000

All measures

Method	Rank
BagMultiRandBal	3.7772
OVA-BagRandBal	3.9936
OVA-SMOTEBagging	4.3157
OverMultiRoughBalBag	5.4567
SMOTEBagging	5.4856
OVO-SMOTEBagging	5.7404
OVA-RoughBalBag	6.0144
OVO-BagRandBal	6.1619
UnderMultiRoughBalBag	6.4503
OVO-RoughBalBag	7.6042

G-mean

Method	Rank	$p_{Hochberg}$
OverMultiRoughBalBag	4.1346	
BagMultiRandBal	4.7885	0.270820
UnderMultiRoughBalBag	4.8654	0.270820
OVA-SMOTEBagging	5.0865	0.270820
OVA-RoughBalBag	5.1442	0.270820
SMOTEBagging	5.2596	0.270820
OVO-RoughBalBag	5.4808	0.140295
OVO-SMOTEBagging	6.2404	0.002733
OVA-BagRandBal	6.2596	0.002733
OVO-BagRandBal	7.7404	0.000000

average-Accuracy

Method	Rank	$p_{Hochberg}$
BagMultiRandBal	3.8462	
OverMultiRoughBalBag	4.2500	0.496417
OVA-SMOTEBagging	4.8846	0.160608
UnderMultiRoughBalBag	5.1923	0.070148
OVA-RoughBalBag	5.2308	0.070148
SMOTEBagging	5.5385	0.021853
OVA-BagRandBal	5.8077	0.005729
OVO-SMOTEBagging	6.1731	0.000623
OVO-RoughBalBag	6.5769	0.000034
OVO-BagRandBal	7.5000	0.000000

F-measure

Method	Rank	$p_{Hochberg}$
BagMultiRandBal	3.5192	
OVA-SMOTEBagging	3.5577	0.948353
OVA-BagRandBal	4.1923	0.513956
OVO-SMOTEBagging	5.0769	0.026118
SMOTEBagging	5.1731	0.021390
OverMultiRoughBalBag	5.5769	0.002647
OVO-BagRandBal	6.1346	0.000064
OVA-RoughBalBag	6.5192	0.000003
UnderMultiRoughBalBag	7.2115	0.000000
OVO-RoughBalBag	8.0385	0.000000

MAUC

Method	Rank	$p_{Hochberg}$
BagMultiRandBal	2.5769	
OVA-BagRandBal	3.0385	0.436982
UnderMultiRoughBalBag	5.2885	0.000005
OverMultiRoughBalBag	5.4808	0.000001
OVA-RoughBalBag	5.5192	0.000001
SMOTEBagging	5.9808	0.000000
OVA-SMOTEBagging	6.1154	0.000000
OVO-BagRandBal	6.1538	0.000000
OVO-SMOTEBagging	6.9615	0.000000

are noticeably lower than the bars in the right subplot, which demonstrates the overall lower ranks of the ensemble methods using Random Balance.

Figure 2 shows the ranks as boxplots. The statistics are calculated across the 52 data sets. The average rank for all methods spans the interval from 1 to 25. The order of the methods is from Table 4, for the subtable with all measures. The boxplots for the methods with Random Balance are coloured in grey. The advantage of the methods with Random Balance is evident from the positioning of their boxes towards the left edge, indicating lower ranks.

Scatterplots of the points for the 52 data sets for the 6 measures are shown in Figure 3. The x-coordinate of a point is the average of the measure for all methods which do not use Random Balance for the corresponding data set, and the y-axis is the average of the measure for the methods which do use Random Balance. If the methods with and without Random Balance would give the same value of a measure for a given data set, the point would lie on the diagonal shown in the plot. The figure shows that all measures apart from the Geometric mean clearly favour the ensemble methods which use Random Balance.

Table 7 shows pair-wise comparisons of the 25 methods. The value in cell (i, j) is the number of data sets where method j has a better result than method i. Table 8 has the same structure and appearance as Table 7, but this time the value in cell (i, j) is the statistically significant wins of method j against method i, according to the *corrected resampled t-test statistic* [70]. For all the measures, in Tables 7 and 8 methods

Table 6: Average ranks for the Boosting-based ensemble methods. Rows for methods with Random Balance are highlighted.

Accuracy

Method	Rank	$p_{Hochberg}$
OVA-RandBalBoost	2.3462	
MultiRandBalBoost	3.7500	0.047107
OVO-RandBalBoost	3.8365	0.047107
OVA-SMOTE+AdaBoost	4.4423	0.009098
OVO-SMOTE+AdaBoost	5.0769	0.000450
OVA-RUSBoost	5.2885	0.000158
OVA-EasyEnsemble	6.9519	0.000000
OVO-RUSBoost	7.3365	0.000000
OVO-EasyEnsemble	9.0385	0.000000
OVA-AdaBoost.NC	9.2692	0.000000
AdaBoost.NC	10.1250	0.000000
OVO-AdaBoost.NC	10.5385	0.000000

Kappa

Method	Rank	$p_{Hochberg}$
OVA-RandBalBoost	2.8654	
MultiRandBalBoost	3.8077	0.182655
OVO-RandBalBoost	4.1346	0.145319
OVA-SMOTE+AdaBoost	5.0962	0.004819
OVA-RUSBoost	5.1346	0.004819
OVO-SMOTE+AdaBoost	5.1731	0.004819
OVA-EasyEnsemble	6.2885	0.000008
OVO-RUSBoost	7.1346	0.000000
OVO-EasyEnsemble	8.4231	0.000000
OVA-AdaBoost.NC	9.5769	0.000000
AdaBoost.NC	9.8654	0.000000
OVO-AdaBoost.NC	10.5000	0.000000

All measures

Method	Rank
OVA-RandBalBoost	3.8141
MultiRandBalBoost	3.9936
OVO-RandBalBoost	4.9439
OVA-RUSBoost	5.0817
OVA-SMOTE+AdaBoost	5.7019
OVO-SMOTE+AdaBoost	6.0673
OVA-EasyEnsemble	6.1474
OVO-RUSBoost	7.1747
OVO-EasyEnsemble	7.3782
AdaBoost.NC	8.6554
OVA-AdaBoost.NC	9.2051
OVO-AdaBoost.NC	9.8365

G-mean

Method	Rank	$p_{Hochberg}$
OVA-RUSBoost	5.0481	
MultiRandBalBoost	5.0962	0.945793
OVA-EasyEnsemble	5.3558	0.945793
OVO-EasyEnsemble	5.4038	0.945793
OVA-RandBalBoost	5.6538	0.945793
OVO-SMOTE+AdaBoost	6.2500	0.445865
OVO-RandBalBoost	6.4231	0.310980
AdaBoost.NC	6.6346	0.173960
OVA-SMOTE+AdaBoost	6.8269	0.095046
OVO-RUSBoost	7.4423	0.006384
OVO-AdaBoost.NC	8.3462	0.000031
OVA-AdaBoost.NC	9.5192	0.000000

average-Accuracy

Method	Rank	$p_{Hochberg}$
MultiRandBalBoost	4.5192	
OVA-RandBalBoost	5.0962	0.414562
OVA-RUSBoost	5.1346	0.414562
OVA-EasyEnsemble	5.2885	0.414562
OVO-RandBalBoost	5.7500	0.327036
OVO-EasyEnsemble	5.8462	0.302897
OVO-SMOTE+AdaBoost	6.2500	0.086268
OVO-RUSBoost	6.7692	0.010239
OVA-SMOTE+AdaBoost	6.8846	0.006579
AdaBoost.NC	7.3654	0.000513
OVO-AdaBoost.NC	9.3462	0.000000
OVA-AdaBoost.NC	9.7500	0.000000

F-measure

Method	Rank	$p_{Hochberg}$
MultiRandBalBoost	3.6731	
OVA-RandBalBoost	3.6923	0.978303
OVO-RandBalBoost	4.7115	0.283879
OVA-RUSBoost	4.9231	0.231300
OVA-SMOTE+AdaBoost	5.0769	0.188427
OVO-SMOTE+AdaBoost	5.2115	0.147884
OVA-EasyEnsemble	6.7885	0.000063
OVO-RUSBoost	7.1346	0.000007
OVO-EasyEnsemble	8.0769	0.000000
AdaBoost.NC	9.0000	0.000000
OVA-AdaBoost.NC	9.6346	0.000000
OVO-AdaBoost.NC	10.0769	0.000000

MAUC

Method	Rank	$p_{Hochberg}$
MultiRandBalBoost	3.1154	
OVA-RandBalBoost	3.2308	0.870378
OVO-RandBalBoost	4.8077	0.033397
OVA-RUSBoost	4.9615	0.027095
OVA-SMOTE+AdaBoost	5.8846	0.000360
OVA-EasyEnsemble	6.2115	0.000060
OVO-RUSBoost	7.2308	0.000000
OVA-AdaBoost.NC	7.4808	0.000000
OVO-EasyEnsemble	7.4808	0.000000
OVO-SMOTE+AdaBoost	8.4423	0.000000
AdaBoost.NC	8.9423	0.000000
OVO-AdaBoost.NC	10.2115	0.000000

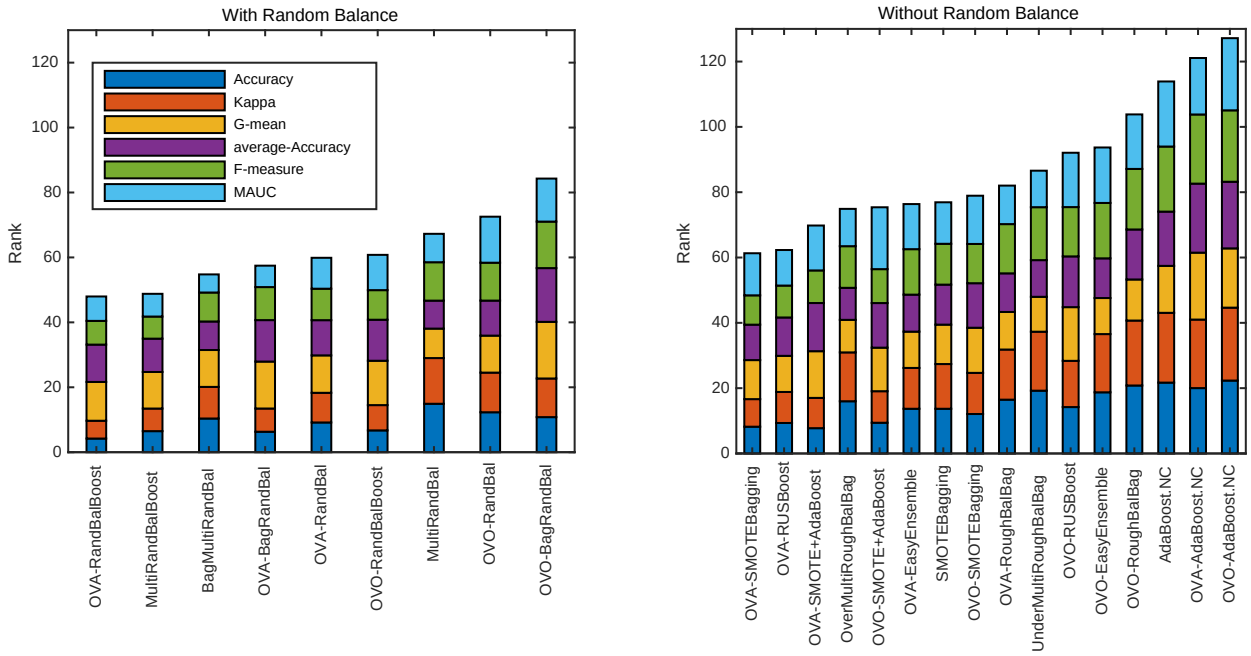


Figure 1: Stacked bar chart of the ensemble methods' ranks according to the six measures, with and without Random Balance.

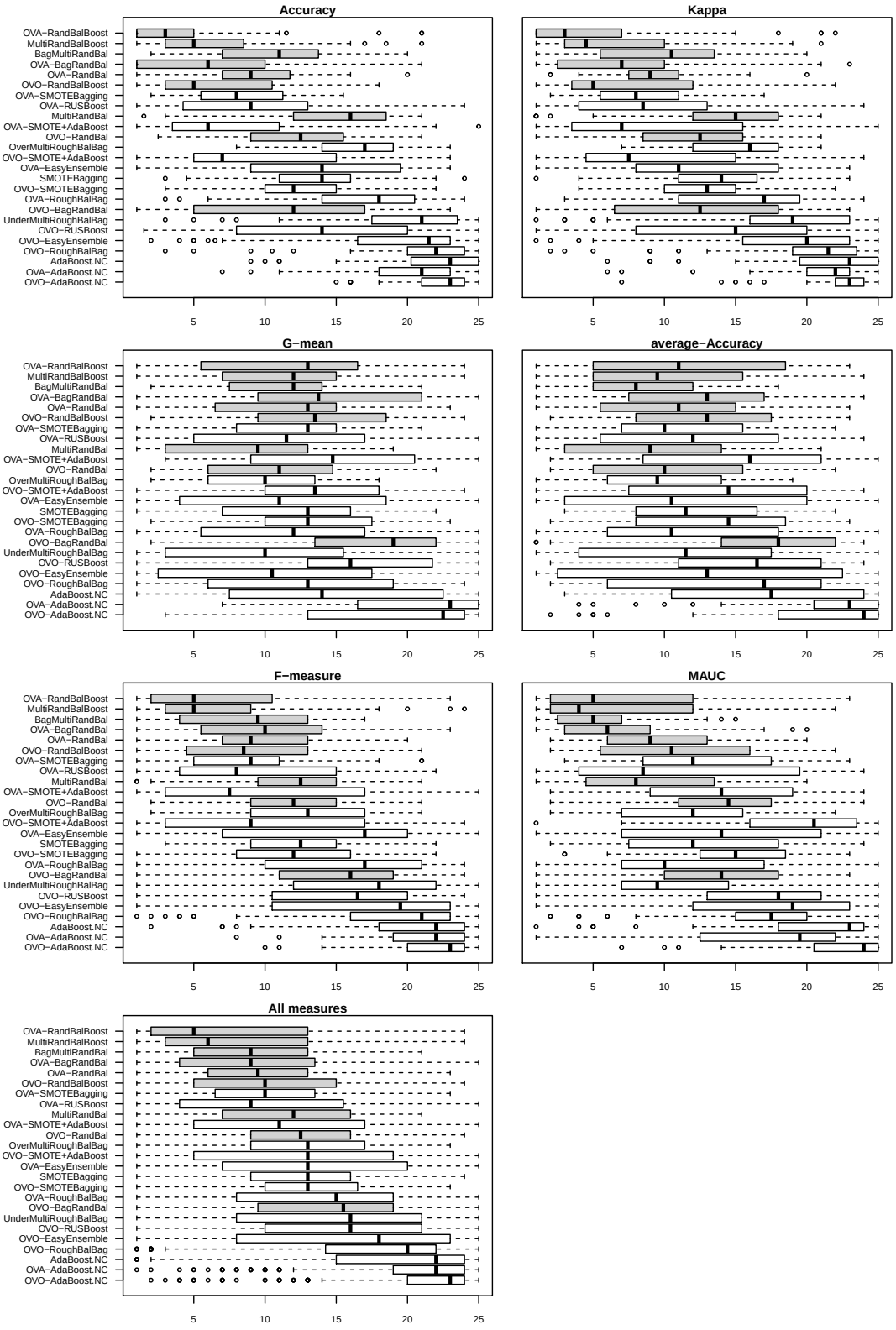


Figure 2: Boxplots for the ranks. The start and end of the box are the first and third quartiles, the band inside the box is the median. The boxplots for the methods with Random Balance are coloured in grey.

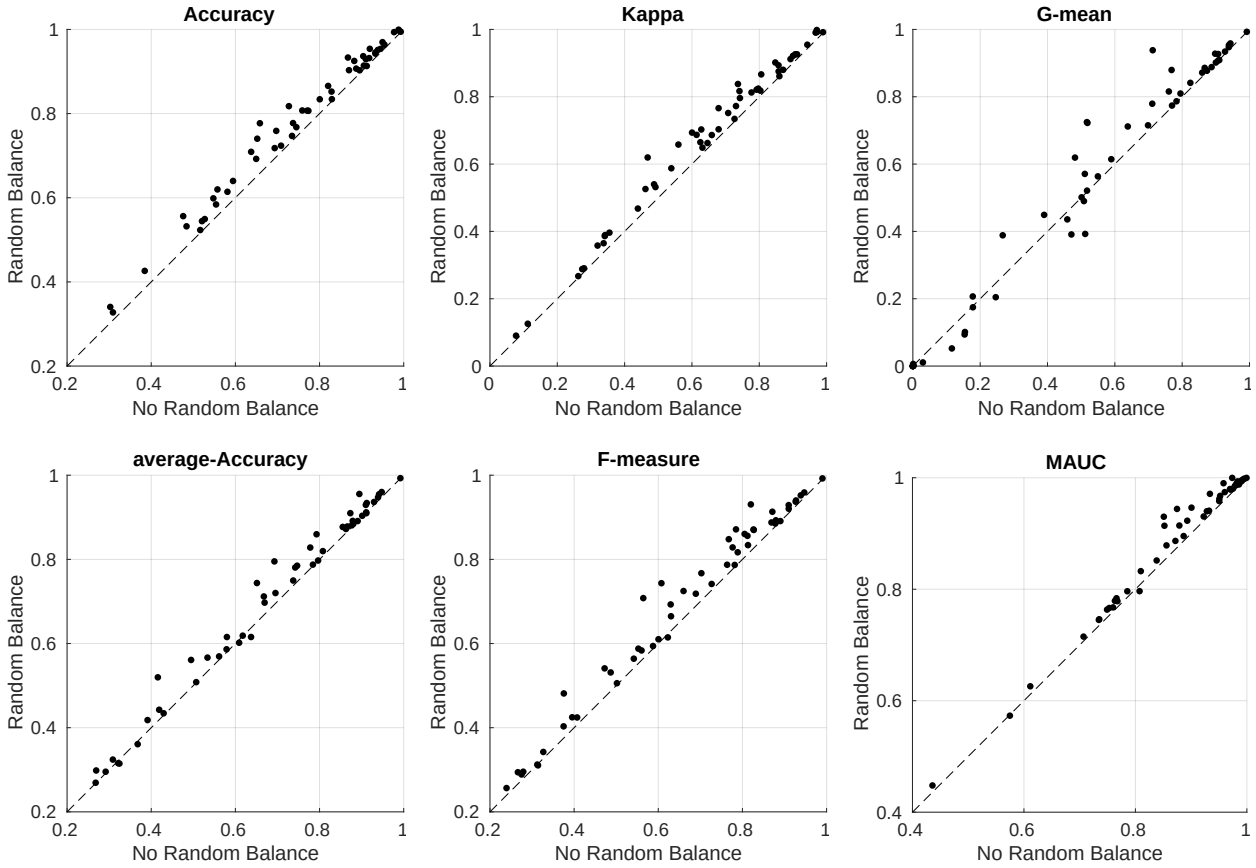


Figure 3: Scatterplot of the points for the 52 data sets for the 6 measures.

with Random Balance have more favourable results.

Table 9 compares the decomposition techniques for the different ensemble methods. The differences between OVA and OVO are more clearly visible here than in the tables with the average ranks. Our experiment showed that, in general, OVA dominates OVO. From the ensemble methods, the greatest differences are for BagRandBal and RoughBalBag, while the smallest differences are for SMOTE+AdaBoost (with the exception of MAUC). Among the performance measures, accuracy and MAUC favours OVA with a largest difference, while G-mean and average-Accuracy show more balanced scores for the two decomposition techniques.

5.1. Decomposition of MAUC

MAUC is calculated as the averaged AUC from all pairs of classes. A pair of classes can be divided in three groups: 1) both classes are majority classes, 2) both are minority, and 3) one class is majority and the other minority. The average AUC can be calculated using only the pairs of each group, resulting in three measures: MAUC-Maj, MAUC-Min and MAUC-Imb. It is possible that some of the groups may be empty. For imbalanced data with only one big class, MAUC-Maj will be empty; for data with only one small class, MAUC-Min will be empty. For those cases we assign a value of 0.5.

Table 10 shows the average ranks for these different versions of MAUC. The order of the methods for the average ranks according to MAUC and MAUC-Imb are very similar.

Figure 4 shows scatter plots for the MAUC measures, representing each data set as a dot, for eight methods. These methods were selected from the top positions in the subtables of Table 4, four with Random Balance and the other four without. MAUC-Imb is the most similar to MAUC. For MAUC-Min, in the majority of the cases its value is smaller than the value of MAUC. For MAUC-Maj and MAUC-Min there are several data sets where there are not pair of classes in the corresponding group, so there are several dots with a value of 0.5 for that measure.

Given that MAUC is the average of the AUC for each pair of classes, it may seem that OVO is more adequate than OVA because it also works with pairs of classes. Nevertheless, the MAUC is not directly calculated from the binary classifiers of class pairs. The predictions of the binary classifiers are combined thereby obtaining a probability for each class. Then, these probabilities are used to calculate the MAUC. When combining the probabilities of the binary classifiers, one issue is that many of the binary classifiers will be necessarily wrong, because they discriminate between two classes and the actual class can be another one. Hence, the probabilities assigned by OVO are not reliable. For other performance measures the results can be good as long as the correct class has the greatest probability, but for MAUC the probabilities assigned to all the classes are considered.

5.2. Bayesian tests

Tables 11 and 12 show the results from the Bayesian signed-rank tests. From all the pairs of methods, the tables only show a subset, the selected set of eight methods compared with all the rest. For all the

Table 7: Pair-wise method comparison. Each cell shows the number of data sets where the method in the column has a better score on the measure of the subtable than the method of the row. Cells background colours are used to represent their values, the better methods have lighter rows and darker columns.

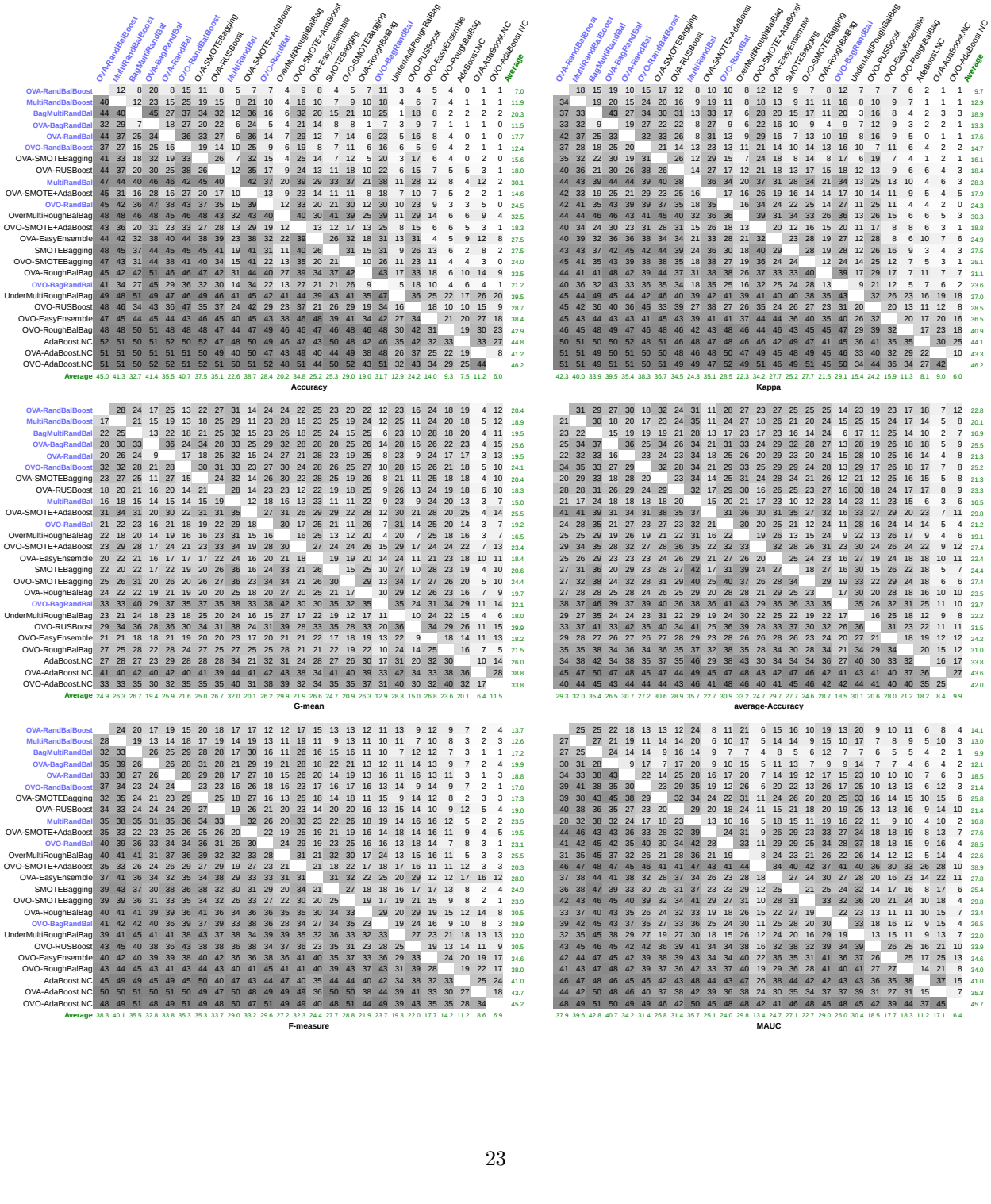
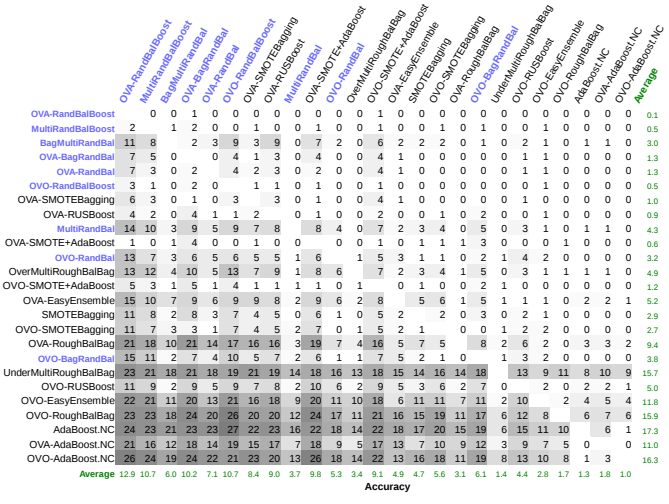
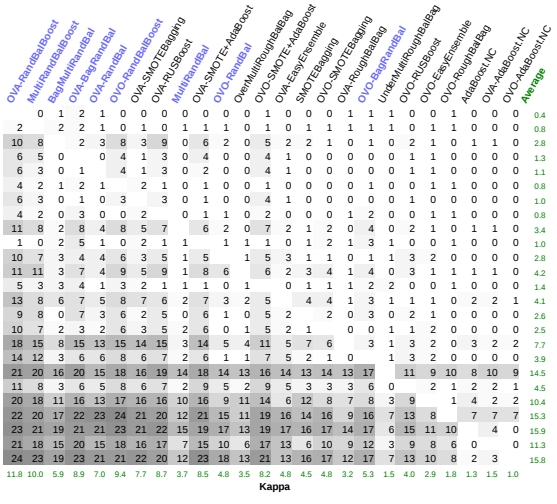
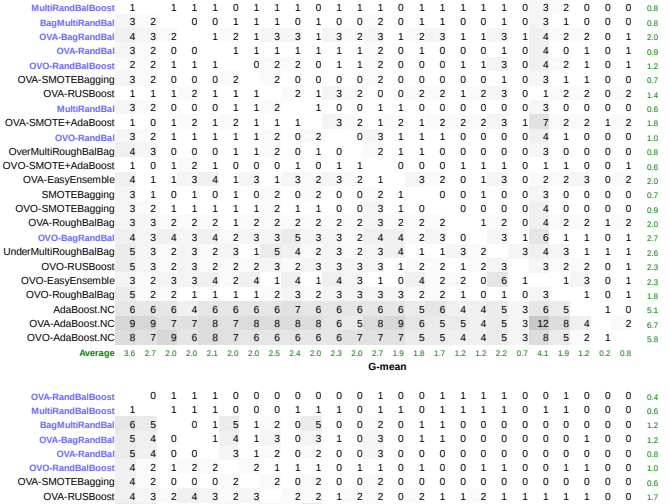
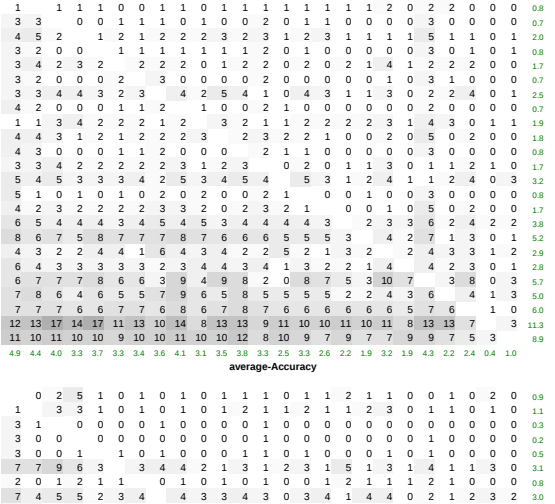


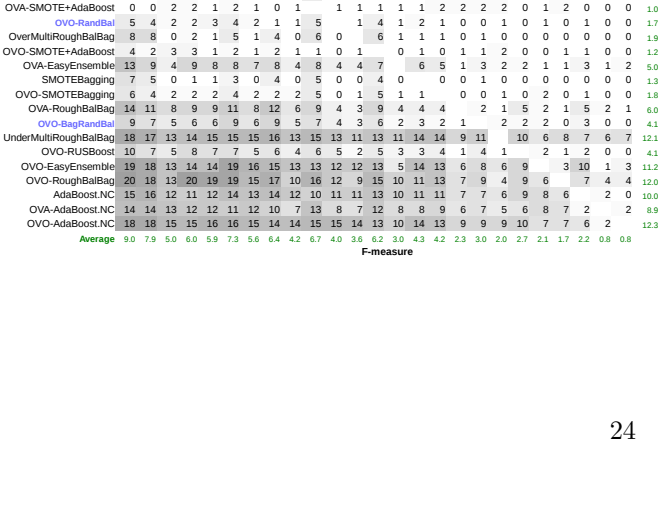
Table 8: Pair-wise method comparison. Each cell shows the number of statistically significant wins across the 52 data sets of the method of the column against the method of the row.











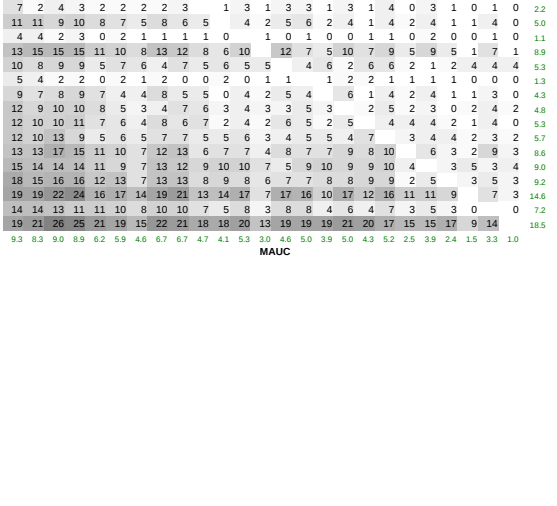


Table 9: Comparison of OVA and OVO for the different ensemble methods and performance measures. Each entry in the table shows the number of data sets where OVA is better followed by the number where OVO is better. The second row for each performance measure shows in how many data sets the differences are significant.

Measure	SMOTEBagging	RoughBalBag	EasyEnsemble	SMOTE+AdaBoost	RUSBoost	AdaBoost.NC	RandBal	BagRandBal	RandBalBoost	**SUM**
Accuracy	40/12 4/0	46/6 11/0	48/4 6/1	29/23 1/1	37/15 8/0	44/8 3/0	38/14 5/0	45/7 7/0	37/15 3/0	364/104 48/2
Kappa	38/14 3/0	45/7 9/0	44/8 6/1	26/26 1/1	39/13 7/0	42/10 3/0	39/13 4/0	33/9 6/0	37/15 4/0	343/115 43/2
G-mean	26/19 1/0	22/23 1/2	22/21 0/2	19/26 0/1	31/13 3/0	17/28 1/2	21/24 1/1	29/14 3/1	32/13 2/1	219/181 12/10
average- Accuracy	31/21 2/0	34/18 2/2	28/24 0/1	22/30 1/1	34/18 2/0	25/27 3/3	27/25 2/1	37/13 8/1	34/18 3/1	272/194 23/10
F-measure	34/18 2/0	37/15 7/1	40/12 5/1	27/25 1/1	38/14 6/1	34/18 2/2	34/18 3/0	40/12 6/0	37/15 4/0	321/147 36/6
MAUC	32/20 4/1	41/11 8/0	36/16 5/1	43/9 8/1	39/13 12/0	45/7 14/0	35/17 8/0	37/9 7/0	39/13 7/0	347/115 73/3
SUM	201/104 16/1	225/80 38/5	218/85 22/7	166/139 12/6	218/86 38/1	207/98 26/7	194/111 23/2	221/64 37/2	216/89 23/2	

Table 10: Average ranks for the MAUC measures. On the right side it is indicated if the method uses OVA or OVO.

	MAUC	MAUC-Maj	MAUC-Min	MAUC-Imb		
BagMultiRandBal	5.577	9.183	7.029	5.500		
OVA-BagRandBal	6.596	7.712	11.673	6.433	OVA	
MultiRandBalBoost	7.000	10.567	8.702	7.269		
OVA-RandBalBoost	7.519	9.529	10.154	7.404	OVA	
MultiRandBal	8.750	12.067	8.144	8.731		
OVA-RandBal	9.519	10.029	13.404	9.183	OVA	
OVO-RandBalBoost	10.865	13.404	10.933	10.375		OVO
OVA-RUSBoost	10.885	10.808	14.356	10.500	OVA	
UnderMultiRoughBalBag	11.173	14.058	7.510	11.231		
OverMultiRoughBalBag	11.423	12.356	11.731	11.481		
OVA-RoughBalBag	11.789	10.125	15.404	11.414	OVA	
SMOTEBagging	12.712	12.808	13.779	13.135		
OVA-SMOTEBagging	12.885	10.462	16.769	12.577	OVA	
OVO-BagRandBal	13.250	14.048	11.058	12.712		OVO
OVA-SMOTE+AdaBoost	13.750	12.837	15.894	14.327	OVA	
OVA-EasyEnsemble	13.808	9.760	16.567	13.885	OVA	
OVO-RandBal	14.173	15.635	11.548	13.990		OVO
OVO-SMOTEBagging	14.769	14.721	12.519	14.606		OVO
OVO-RUSBoost	16.635	13.577	13.587	16.885		OVO
OVO-RoughBalBag	16.692	14.654	12.519	16.875		OVO
OVO-EasyEnsemble	16.981	14.260	14.817	17.414		OVO
OVA-AdaBoost.NC	17.289	16.106	17.269	17.865	OVA	
OVO-SMOTE+AdaBoost	18.942	16.875	15.587	18.914		OVO
AdaBoost.NC	19.923	19.269	15.317	20.183		
OVO-AdaBoost.NC	22.096	20.154	18.731	22.115		OVO

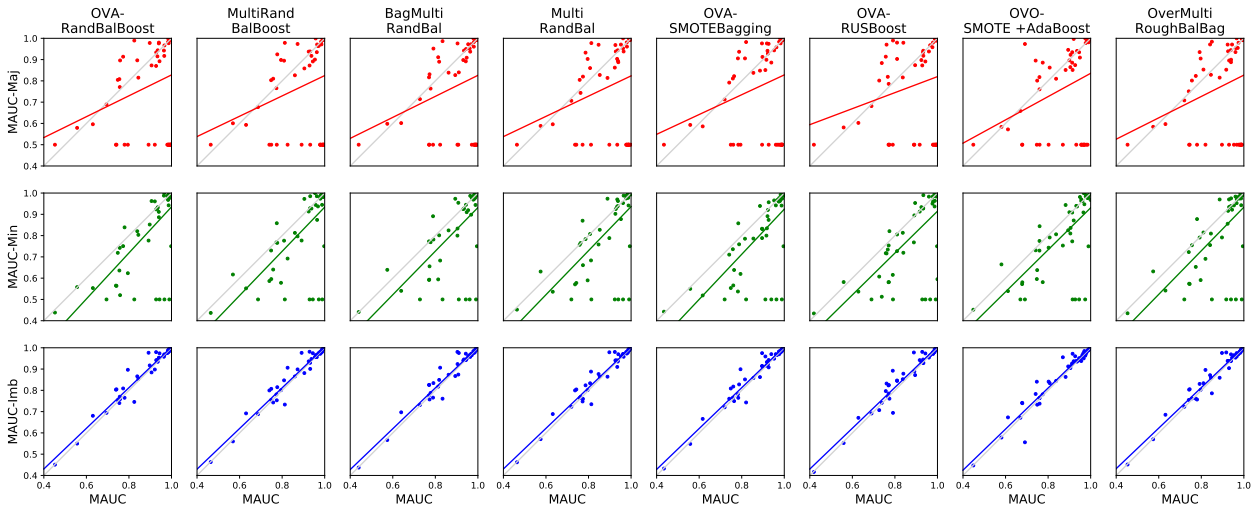


Figure 4: Scatter plots for the MAUC measures. Each dot in a plot is for one data set. The plots show the regression and diagonal lines.

measures, from the eight selected methods, the method with most favourable results in the test is one with Random Balance. Moreover, for all the measures, one of the four selected methods with Random Balance its probability of being better than the other method is greater or equal than the probability of being worse.

Figures from 5 to 10 show the posteriors for the Bayesian sign-rank tests. In these triangles [67], the bottom-left and bottom-right regions correspond to the case where one method is better than the other or vice versa. The top region represents the case where the "rope" is more probable. For each measure, the four selected methods with Random Balance are compared with the four selected methods without Random Balance. Thus, there are 4×4 triangles for each measure. Methods with Random Balance are in the rows while methods without are in the columns. The bottom left region of the triangles are for the case where the method with Random Balance is better. For all the measures, there is a method with Random Balance for which its four triangles (in a row) have their points clouds closer to the left than to the right of the triangle. On the other hand, for all the measures, there is not a method without Random Balance for which its four triangles (in a column) have their points closer to the right than to the left.

5.3. Diversity

The advantage of some ensemble methods can be due to the additional diversity in the base classifiers. There are several measures of diversity [19], one of which is Kappa. When this measure is used as a performance measure, the predicted classes are compared to the actual classes. When Kappa is used for measuring diversity between a pair of base classifiers, the overall diversity measure is the average of the pair-wise values of Kappa from all the pairs. Smaller values of Kappa indicate that the base classifiers are more diverse.

When using binarization techniques, different base classifiers predict different binary classes. We compute

Table 11: Probabilities for the comparisons of classifiers, obtained using the Bayesian signed-rank test. The three probabilities in each cell are for: column method is better / "rope" / row method is better. Continues on Table 12.

Accuracy

	OVA-Rand BalBoost	MultiRand BalBoost	BagMulti RandBal	Multi RandBal	OVA-SMOTE Bagging	OVA-RUSBoost	OVA-SMOTE +AdaBoost	OverMulti RoughBalBag
OVA-RandBalBoost		0.00/1.00/0.00	0.00/0.08/0.92	0.00/0.00/1.00	0.00/0.97/0.03	0.00/0.45/0.55	0.00/0.96/0.04	0.00/0.00/1.00
MultiRandBalBoost	0.00/1.00/0.00		0.00/0.67/0.33	0.00/0.00/1.00	0.00/1.00/0.00	0.00/0.99/0.01	0.00/1.00/0.00	0.00/0.00/1.00
BagMultiRandBal	0.92/0.08/0.00	0.33/0.67/0.00		0.00/0.36/0.64	0.00/1.00/0.00	0.04/0.93/0.02	0.31/0.69/0.00	0.00/0.10/0.90
OVA-BagRandBal	0.05/0.95/0.00	0.00/0.99/0.01	0.00/0.59/0.41	0.00/0.00/1.00	0.00/0.99/0.01	0.00/0.29/0.71	0.00/0.91/0.09	0.00/0.00/1.00
OVA-RandBal	0.67/0.33/0.00	0.04/0.96/0.00	0.00/1.00/0.00	0.00/0.16/0.84	0.00/1.00/0.00	0.00/0.99/0.01	0.01/0.99/0.00	0.00/0.05/0.95
OVO-RandBalBoost	0.02/0.98/0.00	0.00/1.00/0.00	0.00/0.55/0.45	0.00/0.00/1.00	0.00/1.00/0.00	0.00/0.99/0.01	0.00/1.00/0.00	0.00/0.00/1.00
OVA-SMOTEBagging	0.03/0.97/0.00	0.00/1.00/0.00	0.00/1.00/0.00	0.00/0.02/0.98		0.00/0.94/0.06	0.00/1.00/0.00	0.00/0.00/1.00
OVA-RUSBoost	0.55/0.45/0.00	0.01/0.99/0.00	0.02/0.93/0.04	0.00/0.05/0.95	0.05/0.95/0.00		0.01/0.99/0.00	0.00/0.00/1.00
MultiRandBal	1.00/0.00/0.00	1.00/0.00/0.00	0.64/0.36/0.00		0.98/0.02/0.00	0.95/0.05/0.00	1.00/0.00/0.00	0.00/1.00/0.00
OVA-SMOTE+AdaBoost	0.04/0.96/0.00	0.00/1.00/0.00	0.00/0.70/0.30	0.00/0.00/1.00	0.00/1.00/0.00	0.00/0.99/0.01		0.00/0.00/1.00
OVO-RandBal	1.00/0.00/0.00	0.96/0.04/0.00	0.01/0.99/0.00	0.00/0.89/0.11	0.45/0.55/0.00	0.12/0.88/0.00	0.89/0.11/0.00	0.00/0.86/0.14
OverMultiRoughBalBag	1.00/0.00/0.00	1.00/0.00/0.00	0.90/0.10/0.00	0.00/1.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	
OVO-SMOTE+AdaBoost	0.24/0.76/0.00	0.00/1.00/0.00	0.01/0.94/0.04	0.00/0.00/1.00	0.00/0.99/0.00	0.00/1.00/0.00	0.00/1.00/0.00	0.00/0.00/1.00
OVA-EasyEnsemble	1.00/0.00/0.00	1.00/0.00/0.00	0.85/0.15/0.00	0.04/0.96/0.01	0.97/0.03/0.00	0.93/0.07/0.00	0.99/0.01/0.00	0.09/0.83/0.08
SMOTEBagging	1.00/0.00/0.00	1.00/0.00/0.00	0.16/0.84/0.00	0.00/1.00/0.00	0.88/0.12/0.00	0.76/0.24/0.00	0.97/0.03/0.00	0.00/1.00/0.00
OVO-SMOTEBagging	1.00/0.00/0.00	0.83/0.17/0.00	0.00/1.00/0.00	0.00/0.52/0.48	0.18/0.82/0.00	0.13/0.87/0.00	0.84/0.16/0.00	0.00/0.47/0.53
OVA-RoughBalBag	1.00/0.00/0.00	1.00/0.00/0.00	0.96/0.04/0.00	0.62/0.38/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	0.42/0.58/0.00
OVO-BagRandBal	0.98/0.02/0.00	0.75/0.24/0.00	0.00/1.00/0.00	0.01/0.09/0.90	0.17/0.83/0.00	0.32/0.63/0.05	0.53/0.47/0.00	0.00/0.03/0.97
UnderMultiRoughBalBag	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00
OVO-RUSBoost	1.00/0.00/0.00	1.00/0.00/0.00	0.90/0.10/0.00	0.34/0.64/0.02	0.98/0.02/0.00	0.64/0.36/0.00	0.99/0.01/0.00	0.27/0.70/0.03
OVO-EasyEnsemble	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00
OVO-RoughBalBag	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00
AdaBoost.NC	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00
OVA-AdaBoost.NC	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00
OVO-AdaBoost.NC	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00
MEAN	0.73/0.27/0.00	0.62/0.38/0.00	0.44/0.46/0.10	0.29/0.25/0.45	0.53/0.47/0.00	0.50/0.45/0.06	0.61/0.39/0.01	0.28/0.23/0.48

Kappa

	OVA-Rand BalBoost	MultiRand BalBoost	BagMulti RandBal	Multi RandBal	OVA-SMOTE Bagging	OVA-RUSBoost	OVA-SMOTE +AdaBoost	OverMulti RoughBalBag
OVA-RandBalBoost		0.00/1.00/0.00	0.00/0.01/0.99	0.00/0.00/1.00	0.00/0.54/0.46	0.00/0.38/0.62	0.00/0.30/0.70	0.00/0.00/1.00
MultiRandBalBoost	0.00/1.00/0.00		0.00/0.26/0.74	0.00/0.00/1.00	0.00/0.97/0.03	0.00/0.93/0.07	0.00/0.90/0.10	0.00/0.00/1.00
BagMultiRandBal	0.99/0.01/0.00	0.74/0.26/0.00		0.00/0.37/0.63	0.05/0.95/0.00	0.58/0.25/0.16	0.72/0.08/0.20	0.00/0.14/0.86
OVA-BagRandBal	0.65/0.32/0.03	0.17/0.73/0.10	0.00/0.46/0.54	0.00/0.00/1.00	0.00/0.98/0.02	0.02/0.22/0.76	0.05/0.22/0.73	0.00/0.00/1.00
OVA-RandBal	0.95/0.05/0.00	0.37/0.62/0.00	0.00/0.99/0.01	0.00/0.09/0.91	0.00/1.00/0.00	0.07/0.65/0.28	0.17/0.57/0.26	0.00/0.02/0.98
OVO-RandBalBoost	0.39/0.61/0.00	0.01/0.99/0.00	0.02/0.21/0.77	0.00/0.00/1.00	0.06/0.89/0.05	0.00/0.98/0.02	0.01/0.95/0.04	0.00/0.00/1.00
OVA-SMOTEBagging	0.45/0.55/0.00	0.03/0.97/0.00	0.00/0.95/0.05	0.00/0.02/0.98		0.03/0.55/0.42	0.02/0.77/0.21	0.00/0.00/1.00
OVA-RUSBoost	0.62/0.38/0.00	0.07/0.93/0.00	0.16/0.25/0.59	0.00/0.00/1.00	0.42/0.55/0.03		0.00/0.99/0.00	0.00/0.00/1.00
MultiRandBal	1.00/0.00/0.00	1.00/0.00/0.00	0.63/0.37/0.00		0.98/0.02/0.00	1.00/0.00/0.00	0.99/0.00/0.00	0.00/1.00/0.00
OVA-SMOTE+AdaBoost	0.70/0.30/0.00	0.10/0.90/0.00	0.20/0.08/0.72	0.00/0.00/0.99	0.21/0.77/0.02	0.00/0.99/0.00		0.00/0.00/1.00
OVO-RandBal	1.00/0.00/0.00	1.00/0.00/0.00	0.08/0.92/0.00	0.00/0.95/0.05	0.75/0.25/0.00	0.91/0.08/0.00	0.98/0.01/0.01	0.00/0.90/0.10
OverMultiRoughBalBag	1.00/0.00/0.00	1.00/0.00/0.00	0.86/0.14/0.00	0.00/1.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	
OVO-SMOTE+AdaBoost	0.67/0.33/0.00	0.03/0.97/0.00	0.31/0.22/0.47	0.00/0.00/1.00	0.14/0.84/0.02	0.02/0.98/0.00	0.01/0.99/0.00	0.00/0.00/1.00
OVA-EasyEnsemble	1.00/0.00/0.00	0.99/0.01/0.00	0.87/0.13/0.00	0.18/0.61/0.21	0.91/0.09/0.00	0.89/0.11/0.00	0.95/0.04/0.00	0.12/0.49/0.39
SMOTEBagging	1.00/0.00/0.00	1.00/0.00/0.00	0.74/0.26/0.00	0.00/1.00/0.00	0.98/0.02/0.00	0.99/0.01/0.00	0.99/0.01/0.00	0.00/1.00/0.00
OVO-SMOTEBagging	1.00/0.00/0.00	0.99/0.01/0.00	0.09/0.91/0.00	0.00/0.87/0.13	0.73/0.27/0.00	0.89/0.11/0.00	0.99/0.01/0.00	0.00/0.80/0.20
OVA-RoughBalBag	1.00/0.00/0.00	1.00/0.00/0.00	0.99/0.01/0.00	0.77/0.23/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	0.47/0.52/0.02
OVO-BagRandBal	1.00/0.00/0.00	1.00/0.00/0.00	0.09/0.91/0.00	0.13/0.09/0.77	0.88/0.12/0.00	0.98/0.02/0.01	0.99/0.00/0.01	0.05/0.05/0.90
UnderMultiRoughBalBag	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00
OVO-RUSBoost	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	0.88/0.10/0.01	1.00/0.00/0.00	0.91/0.09/0.00	1.00/0.00/0.00	0.79/0.14/0.07
OVO-EasyEnsemble	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00
OVO-RoughBalBag	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00
AdaBoost.NC	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00
OVA-AdaBoost.NC	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00
OVO-AdaBoost.NC	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00
MEAN	0.85/0.15/0.00	0.69/0.31/0.00	0.50/0.30/0.20	0.33/0.22/0.45	0.63/0.34/0.03	0.64/0.27/0.10	0.66/0.24/0.10	0.31/0.21/0.48

G-mean

	OVA-Rand BalBoost	MultiRand BalBoost	BagMulti RandBal	Multi RandBal	OVA-SMOTE Bagging	OVA-RUSBoost	OVA-SMOTE +AdaBoost	OverMulti RoughBalBag
OVA-RandBalBoost		0.48/0.52/0.00	0.70/0.17/0.13	0.96/0.04/0.01	0.66/0.26/0.08	0.29/0.66/0.05	0.00/0.58/0.42	0.89/0.08/0.03
MultiRandBalBoost	0.00/0.52/0.48		0.03/0.92/0.06	0.93/0.07/0.00	0.01/0.83/0.16	0.07/0.85/0.07	0.00/0.08/0.91	0.68/0.32/0.00
BagMultiRandBal	0.13/0.17/0.71	0.06/0.92/0.03		0.77/0.23/0.00	0.00/0.98/0.01	0.49/0.15/0.36	0.00/0.10/0.90	0.35/0.65/0.00
OVA-BagRandBal	0.58/0.42/0.00	0.99/0.01/0.00	0.91/0.09/0.00	1.00/0.00/0.00	0.96/0.04/0.00	0.97/0.01/0.02	0.30/0.62/0.08	0.97/0.03/0.00
OVA-RandBal	0.20/0.13/0.67	0.30/0.68/0.02	0.09/0.89/0.02	0.96/0.04/0.00	0.01/0.99/0.00	0.39/0.17/0.44	0.00/0.12/0.88	0.48/0.52/0.00
OVO-RandBalBoost	0.44/0.55/0.01	0.63/0.37/0.00	0.73/0.21/0.06	0.99/0.00/0.00	0.85/0.15/0.00	0.70/0.27/0.03	0.07/0.53/0.40	0.94/0.05/0.00
OVA-SMOTEBagging	0.08/0.26/0.66	0.16/0.83/0.01	0.01/0.98/0.00	0.96/0.04/0.00		0.37/0.26/0.37	0.00/0.25/0.75	0.40/0.59/0.00
OVA-RUSBoost	0.05/0.66/0.29	0.07/0.86/0.07	0.36/0.15/0.49	0.76/0.23/0.01	0.37/0.25/0.38		0.01/0.09/0.90	0.80/0.13/0.07
MultiRandBal	0.01/0.03/0.96	0.00/0.07/0.93	0.00/0.23/0.77		0.00/0.04/0.96	0.01/0.23/0.75	0.00/0.00/1.00	0.00/0.95/0.05
OVA-SMOTE+AdaBoost	0.42/0.58/0.00	0.92/0.08/0.00	0.90/0.10/0.00	1.00/0.00/0.00	0.75/0.25/0.00	0.90/0.10/0.01		0.99/0.01/0.00
OVO-RandBal	0.04/0.01/0.95	0.07/0.24/0.69	0.00/0.65/0.35	0.39/0.59/0.02	0.01/0.57/0.42	0.07/0.07/0.87	0.00/0.05/0.95	0.02/0.97/0.00
OverMultiRoughBalBag	0.03/0.08/0.89	0.00/0.32/0.68	0.00/0.65/0.35	0.05/0.95/0.00	0.00/0.60/0.40	0.07/0.13/0.80	0.00/0.01/0.99	
OVO-SMOTE+AdaBoost	0.34/0.63/0.03	0.75/0.25/0.00	0.92/0.07/0.01	1.00/0.00/0.00	0.57/0.29/0.13	0.87/0.13/0.00	0.37/0.39/0.24	0.99/0.01/0.00
OVA-EasyEnsemble	0.66/0.01/0.33	0.79/0.03/0.17	0.85/0.05/0.11	0.95/0.01/0.04	0.79/0.02/0.19	0.64/0.19/0.18	0.42/0.00/0.57	0.92/0.02/0.05
SMOTEBagging	0.09/0.09/0.82	0.00/0.77/0.23	0.00/0.66/0.34	0.68/0.32/0.00	0.01/0.80/0.20	0.40/0.19/0.41	0.00/0.03/0.97	0.02/0.98/0.00
OVO-SMOTEBagging	0.43/0.01/0.56	0.76/0.18/0.06	0.48/0.43/0.08	0.96/0.04/0.00	0.21/0.66/0.13	0.60/0.11/0.29	0.08/0.31/0.61	0.52/0.48/0.00
OVA-RoughBalBag	0.43/0.01/0.56	0.43/0.06/0.52	0.18/0.13/0.69	0.73/0.13/0.14	0.35/0.11/0.54	0.29/0.04/0.68	0.08/0.00/0.92	0.53/0.37/0.10
OVO-BagRandBal	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	0.99/0.01/0.00	1.00/0.00/0.00
UnderMultiRoughBalBag	0.09/0.00/0.91	0.06/0.01/0.94	0.01/0.14/0.85	0.53/0.15/0.32	0.04/0.15/0.81	0.02/0.01/0.98	0.00/0.00/1.00	0.17/0.61/0.21
OVO-RUSBoost	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	0.99/0.01/0.00	1.00/0.00/0.00	0.91/0.03/0.06	1.00/0.00/0.00
OVO-EasyEnsemble	0.62/0.05/0.33	0.63/0.04/0.33	0.69/0.06/0.25	0.85/0.04/0.10	0.66/0.10/0.24	0.43/0.24/0.33	0.40/0.03/0.56	0.75/0.03/0.22
OVO-RoughBalBag	0.71/0.00/0.29	0.73/0.01/0.27	0.60/0.20/0.20	0.92/0.04/0.04	0.77/0.08/0.16	0.56/0.01/0.43	0.52/0.01/0.47	0.70/0.23/0.08
AdaBoost.NC	0.93/0.00/0.07	0.95/0.00/0.05	0.95/0.00/0.05	1.00/0.00/0.00	0.94/0.00/0.05	0.92/0.00/0.08	0.65/0.00/0.35	0.99/0.01/0.00
OVA-AdaBoost.NC	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00
OVO-AdaBoost.NC	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	0.99/0.00/0.01	1.00/0.00/0.00
MEAN	0.43/0.18/0.40	0.53/0.26/0.21	0.52/0.28/0.20	0.85/0.12/0.03	0.50/0.30/0.20	0.54/0.16/0.30	0.28/0.14/0.58	0.67/0.29/0.03

Table 12: Continued from Table 11. Probabilities for the comparisons of classifiers, obtained using the Bayesian signed-rank test. The three probabilities in each cell are for: column method is better / "rope" / row method is better.

Average-Accuracy

	OVA-Rand BalBoost	MultiRand BalBoost	BagMulti RandBal	Multi RandBal	OVA-SMOTE Bagging	OVA-RUSBoost	OVA-SMOTE +AdaBoost	OverMulti RoughBalBag
OVA-RandBalBoost		0.00/1.00/0.00	0.50/0.48/0.01	0.71/0.27/0.02	0.02/0.98/0.00	0.00/0.85/0.15	0.00/0.20/0.80	0.31/0.68/0.02
MultiRandBalBoost	0.00/1.00/0.00		0.02/0.98/0.00	0.07/0.93/0.00	0.00/1.00/0.00	0.00/0.78/0.22	0.00/0.07/0.93	0.01/0.99/0.00
BagMultiRandBal	0.01/0.48/0.51	0.00/0.98/0.02		0.00/1.00/0.00	0.00/0.96/0.04	0.00/0.09/0.90	0.00/0.01/0.99	0.00/1.00/0.00
OVA-BagRandBal	0.29/0.69/0.02	0.72/0.27/0.01	0.57/0.43/0.00	0.88/0.12/0.00	0.09/0.91/0.00	0.11/0.25/0.65	0.01/0.28/0.71	0.64/0.36/0.00
OVA-RandBal	0.05/0.72/0.23	0.04/0.96/0.00	0.01/0.99/0.00	0.11/0.89/0.00	0.00/1.00/0.00	0.01/0.20/0.79	0.00/0.05/0.95	0.03/0.97/0.00
OVO-RandBalBoost	0.08/0.92/0.00	0.12/0.88/0.00	0.68/0.32/0.00	0.65/0.35/0.00	0.29/0.70/0.00	0.01/0.88/0.12	0.01/0.63/0.37	0.37/0.63/0.00
OVA-SMOTEBagging	0.00/0.98/0.02	0.00/1.00/0.00	0.04/0.96/0.00	0.13/0.87/0.00		0.01/0.31/0.68	0.00/0.14/0.86	0.01/0.99/0.00
OVA-RUSBoost	0.15/0.85/0.00	0.23/0.77/0.00	0.91/0.09/0.00	0.82/0.17/0.00	0.68/0.31/0.01		0.05/0.12/0.83	0.62/0.38/0.00
MultiRandBal	0.02/0.27/0.71	0.00/0.93/0.07	0.00/1.00/0.00		0.00/0.87/0.13	0.00/0.18/0.82	0.00/0.00/1.00	0.00/1.00/0.00
OVA-SMOTE+AdaBoost	0.80/0.20/0.00	0.93/0.07/0.00	0.99/0.01/0.00	1.00/0.00/0.00	0.86/0.14/0.00	0.83/0.12/0.06		1.00/0.00/0.00
OVO-RandBal	0.24/0.33/0.43	0.16/0.67/0.16	0.01/0.99/0.00	0.03/0.97/0.00	0.01/0.98/0.01	0.02/0.19/0.78	0.00/0.06/0.94	0.00/1.00/0.00
OverMultiRoughBalBag	0.02/0.67/0.31	0.00/0.99/0.01	0.00/1.00/0.00	0.00/1.00/0.00	0.00/0.99/0.01	0.00/0.38/0.62	0.00/0.01/0.99	
OVO-SMOTE+AdaBoost	0.27/0.73/0.00	0.46/0.54/0.00	0.99/0.01/0.00	0.98/0.02/0.00	0.68/0.31/0.00	0.54/0.45/0.01	0.36/0.55/0.09	0.87/0.13/0.00
OVA-EasyEnsemble	0.97/0.00/0.03	0.99/0.01/0.00	0.99/0.01/0.00	0.99/0.00/0.00	0.98/0.00/0.02	0.88/0.11/0.01	0.69/0.00/0.31	0.98/0.01/0.01
SMOTEBagging	0.04/0.93/0.03	0.00/1.00/0.00	0.02/0.98/0.00	0.09/0.91/0.00	0.00/1.00/0.00	0.08/0.40/0.53	0.00/0.07/0.93	0.00/1.00/0.00
OVO-SMOTEBagging	0.64/0.31/0.06	0.65/0.34/0.01	0.69/0.31/0.00	0.62/0.38/0.00	0.14/0.86/0.00	0.51/0.21/0.28	0.03/0.49/0.48	0.24/0.76/0.00
OVA-RoughBalBag	0.82/0.02/0.16	0.69/0.20/0.11	0.57/0.43/0.00	0.67/0.33/0.00	0.37/0.62/0.01	0.56/0.20/0.23	0.17/0.01/0.82	0.38/0.62/0.00
OVO-BagRandBal	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	0.99/0.01/0.00	0.97/0.00/0.02	1.00/0.00/0.00
UnderMultiRoughBalBag	0.58/0.02/0.40	0.50/0.20/0.29	0.20/0.78/0.02	0.36/0.63/0.01	0.15/0.76/0.09	0.08/0.02/0.90	0.02/0.00/0.98	0.16/0.83/0.01
OVO-RUSBoost	0.99/0.01/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	0.71/0.28/0.01	0.87/0.02/0.11	1.00/0.00/0.00
OVO-EasyEnsemble	0.96/0.00/0.03	0.98/0.00/0.01	0.99/0.01/0.01	0.99/0.00/0.01	0.98/0.00/0.02	0.90/0.06/0.04	0.76/0.00/0.24	0.97/0.00/0.03
OVO-RoughBalBag	1.00/0.00/0.00	1.00/0.00/0.00	0.99/0.01/0.00	0.99/0.01/0.00	1.00/0.00/0.00	1.00/0.00/0.00	0.97/0.00/0.03	0.96/0.04/0.00
AdaBoost.NC	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	0.98/0.00/0.02	0.91/0.00/0.09	1.00/0.00/0.00
OVA-AdaBoost.NC	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00
OVO-AdaBoost.NC	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00
MEAN	0.50/0.38/0.12	0.52/0.45/0.03	0.59/0.41/0.00	0.63/0.37/0.00	0.47/0.52/0.01	0.43/0.25/0.33	0.33/0.11/0.56	0.52/0.47/0.00

F-measure

	OVA-Rand BalBoost	MultiRand BalBoost	BagMulti RandBal	Multi RandBal	OVA-SMOTE Bagging	OVA-RUSBoost	OVA-SMOTE +AdaBoost	OverMulti RoughBalBag
OVA-RandBalBoost		0.00/1.00/0.00	0.00/0.15/0.85	0.00/0.01/0.99	0.00/0.89/0.11	0.00/0.69/0.31	0.00/0.55/0.45	0.00/0.00/1.00
MultiRandBalBoost	0.00/1.00/0.00		0.00/0.70/0.30	0.00/0.03/0.97	0.00/0.96/0.04	0.00/0.76/0.24	0.00/0.62/0.38	0.00/0.00/1.00
BagMultiRandBal	0.85/0.15/0.00	0.31/0.69/0.00		0.00/1.00/0.00	0.01/0.99/0.00	0.11/0.36/0.53	0.33/0.27/0.40	0.00/0.91/0.09
OVA-BagRandBal	0.77/0.23/0.00	0.68/0.32/0.00	0.01/0.99/0.00	0.01/0.83/0.17	0.02/0.98/0.00	0.08/0.47/0.46	0.12/0.52/0.36	0.00/0.85/0.15
OVA-RandBal	0.45/0.55/0.00	0.26/0.74/0.00	0.00/1.00/0.00	0.00/0.95/0.05	0.00/1.00/0.00	0.02/0.63/0.35	0.02/0.70/0.28	0.00/0.74/0.26
OVO-RandBalBoost	0.30/0.70/0.00	0.08/0.92/0.00	0.03/0.79/0.17	0.01/0.22/0.77	0.01/0.99/0.00	0.00/0.99/0.01	0.03/0.90/0.07	0.00/0.16/0.84
OVA-SMOTEBagging	0.11/0.89/0.00	0.04/0.96/0.00	0.00/0.99/0.01	0.00/0.77/0.23		0.01/0.78/0.22	0.01/0.78/0.22	0.00/0.44/0.56
OVA-RUSBoost	0.32/0.68/0.00	0.24/0.76/0.00	0.53/0.36/0.11	0.10/0.59/0.31	0.22/0.77/0.01		0.10/0.83/0.07	0.07/0.31/0.62
MultiRandBal	0.99/0.01/0.00	0.97/0.03/0.00	0.00/1.00/0.00		0.22/0.78/0.00	0.31/0.59/0.10	0.82/0.06/0.11	0.00/1.00/0.00
OVA-SMOTE+AdaBoost	0.45/0.55/0.00	0.38/0.62/0.00	0.40/0.27/0.33	0.11/0.07/0.82	0.21/0.78/0.01	0.07/0.83/0.10		0.03/0.03/0.94
OVO-RandBal	0.99/0.01/0.00	0.97/0.03/0.00	0.14/0.86/0.00	0.00/1.00/0.00	0.30/0.70/0.00	0.32/0.55/0.14	0.81/0.12/0.07	0.00/0.98/0.02
OverMultiRoughBalBag	1.00/0.00/0.00	1.00/0.00/0.00	0.09/0.91/0.00	0.00/1.00/0.00	0.55/0.45/0.00	0.62/0.31/0.07	0.94/0.03/0.03	
OVO-SMOTE+AdaBoost	0.51/0.49/0.00	0.33/0.67/0.00	0.59/0.35/0.06	0.17/0.26/0.56	0.24/0.75/0.00	0.12/0.87/0.00	0.30/0.68/0.01	0.10/0.13/0.77
OVA-EasyEnsemble	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	0.99/0.01/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	0.98/0.02/0.00
SMOTEBagging	0.97/0.03/0.00	0.97/0.03/0.00	0.03/0.97/0.00	0.00/1.00/0.00	0.05/0.95/0.00	0.46/0.37/0.17	0.73/0.23/0.04	0.00/1.00/0.00
OVO-SMOTEBagging	0.99/0.01/0.00	0.99/0.01/0.00	0.22/0.78/0.00	0.04/0.95/0.02	0.28/0.72/0.00	0.68/0.27/0.05	0.95/0.03/0.02	0.01/0.96/0.03
OVA-RoughBalBag	1.00/0.00/0.00	1.00/0.00/0.00	0.99/0.01/0.00	0.97/0.03/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	0.71/0.29/0.00
OVO-BagRandBal	1.00/0.00/0.00	1.00/0.00/0.00	0.91/0.09/0.00	0.59/0.41/0.00	0.99/0.01/0.00	0.99/0.00/0.00	1.00/0.00/0.00	0.44/0.56/0.00
UnderMultiRoughBalBag	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	0.98/0.02/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	0.86/0.14/0.00
OVO-RUSBoost	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	0.98/0.02/0.00	1.00/0.00/0.00	0.99/0.01/0.00	1.00/0.00/0.00	0.94/0.06/0.00
OVO-EasyEnsemble	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00
OVO-RoughBalBag	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00
AdaBoost.NC	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00
OVA-AdaBoost.NC	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00
OVO-AdaBoost.NC	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00
MEAN	0.78/0.22/0.00	0.72/0.28/0.00	0.50/0.43/0.08	0.41/0.38/0.20	0.50/0.49/0.01	0.53/0.35/0.11	0.63/0.26/0.11	0.38/0.36/0.26

MAUC

	OVA-Rand BalBoost	MultiRand BalBoost	BagMulti RandBal	Multi RandBal	OVA-SMOTE Bagging	OVA-RUSBoost	OVA-SMOTE +AdaBoost	OverMulti RoughBalBag
OVA-RandBalBoost		0.00/1.00/0.00	0.00/1.00/0.00	0.00/1.00/0.00	0.00/1.00/0.00	0.00/0.92/0.08	0.00/0.99/0.01	0.00/1.00/0.00
MultiRandBalBoost	0.00/1.00/0.00		0.00/1.00/0.00	0.00/1.00/0.00	0.00/0.97/0.03	0.00/0.82/0.18	0.00/0.93/0.07	0.00/1.00/0.00
BagMultiRandBal	0.00/1.00/0.00	0.00/1.00/0.00		0.00/1.00/0.00	0.00/0.82/0.18	0.00/0.82/0.18	0.00/0.30/0.70	0.00/1.00/0.00
OVA-BagRandBal	0.00/1.00/0.00	0.00/1.00/0.00	0.00/1.00/0.00	0.00/1.00/0.00	0.00/0.98/0.02	0.00/0.83/0.17	0.00/0.63/0.37	0.00/1.00/0.00
OVA-RandBal	0.00/1.00/0.00	0.00/1.00/0.00	0.00/1.00/0.00	0.00/1.00/0.00	0.00/1.00/0.00	0.00/0.94/0.06	0.00/0.99/0.01	0.00/1.00/0.00
OVO-RandBalBoost	0.00/1.00/0.00	0.02/0.98/0.00	0.23/0.77/0.00	0.00/1.00/0.00	0.00/0.99/0.00	0.00/0.99/0.01	0.00/1.00/0.00	0.00/1.00/0.00
OVA-SMOTEBagging	0.00/1.00/0.00	0.03/0.97/0.00	0.18/0.82/0.00	0.00/1.00/0.00		0.00/0.93/0.07	0.00/1.00/0.00	0.00/1.00/0.00
OVA-RUSBoost	0.08/0.92/0.00	0.17/0.83/0.00	0.18/0.82/0.00	0.08/0.92/0.00	0.07/0.93/0.00		0.01/0.99/0.00	0.11/0.89/0.00
MultiRandBal	0.00/1.00/0.00	0.00/1.00/0.00	0.00/1.00/0.00		0.00/1.00/0.00	0.00/0.92/0.08	0.00/0.93/0.07	0.00/1.00/0.00
OVA-SMOTE+AdaBoost	0.01/0.99/0.00	0.07/0.93/0.00	0.71/0.29/0.00	0.07/0.93/0.00	0.00/1.00/0.00	0.00/0.99/0.01		0.02/0.98/0.00
OVO-RandBal	0.19/0.81/0.00	0.68/0.32/0.00	0.62/0.38/0.00	0.18/0.82/0.00	0.00/1.00/0.00	0.01/0.98/0.02	0.00/1.00/0.00	0.01/0.99/0.00
OverMultiRoughBalBag	0.00/1.00/0.00	0.00/1.00/0.00	0.00/1.00/0.00	0.00/1.00/0.00	0.00/1.00/0.00	0.00/0.88/0.12	0.00/0.98/0.02	
OVO-SMOTE+AdaBoost	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	0.79/0.21/0.00	0.97/0.03/0.00	0.61/0.39/0.00	0.99/0.01/0.00
OVA-EasyEnsemble	0.67/0.33/0.00	0.80/0.20/0.00	0.96/0.04/0.00	0.69/0.31/0.00	0.11/0.89/0.00	0.36/0.64/0.00	0.29/0.71/0.00	0.70/0.30/0.00
SMOTEBagging	0.03/0.97/0.00	0.03/0.97/0.00	0.15/0.85/0.00	0.00/1.00/0.00	0.00/1.00/0.00	0.00/0.94/0.06	0.00/1.00/0.00	0.00/1.00/0.00
OVO-SMOTEBagging	0.33/0.67/0.00	0.61/0.39/0.00	0.66/0.34/0.00	0.15/0.85/0.00	0.00/1.00/0.00	0.01/0.97/0.02	0.00/1.00/0.00	0.03/0.97/0.00
OVA-RoughBalBag	0.09/0.91/0.00	0.06/0.94/0.00	0.02/0.98/0.00	0.01/0.99/0.00	0.00/1.00/0.00	0.00/1.00/0.00	0.01/0.97/0.01	0.01/0.99/0.00
OVO-BagRandBal	0.24/0.76/0.00	0.42/0.58/0.00	0.68/0.32/0.00	0.09/0.91/0.00	0.01/0.99/0.00	0.02/0.93/0.05	0.00/1.00/0.00	0.03/0.97/0.00
UnderMultiRoughBalBag	0.05/0.95/0.00	0.08/0.92/0.00	0.06/0.94/0.00	0.01/0.99/0.00	0.00/1.00/0.00	0.04/0.56/0.40	0.01/0.90/0.10	0.00/1.00/0.00
OVO-RUSBoost	0.91/0.09/0.00	0.88/0.12/0.00	0.98/0.02/0.00	0.79/0.21/0.00	0.76/0.24/0.00	0.07/0.93/0.00	0.55/0.45/0.00	0.75/0.25/0.00
OVO-EasyEnsemble	0.96/0.04/0.00	1.00/0.00/0.00	1.00/0.00/0.00	0.99/0.01/0.00	0.83/0.17/0.00	0.84/0.16/0.00	0.76/0.24/0.00	0.94/0.06/0.00
OVO-RoughBalBag	0.92/0.08/0.00	0.97/0.03/0.00	0.97/0.03/0.00	0.76/0.24/0.00	0.37/0.63/0.00	0.07/0.93/0.00	0.21/0.79/0.00	0.63/0.37/0.00
AdaBoost.NC	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00
OVA-AdaBoost.NC	0.98/0.02/0.00	0.97/0.03/0.00	0.99/0.01/0.00	0.95/0.05/0.00	0.33/0.67/0.00	0.63/0.34/0.03	0.21/0.79/0.00	0.83/0.17/0.00
OVO-AdaBoost.NC	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00	1.00/0.00/0.00
MEAN	0.35/0.65/0.00	0.41/0.59/0.00	0.48/0.52/0.00	0.32/0.68/0.00	0.22/0.77/0.01	0.21/0.73/0.06	0.19/0.75/0.06	0.29/0.71/0.00

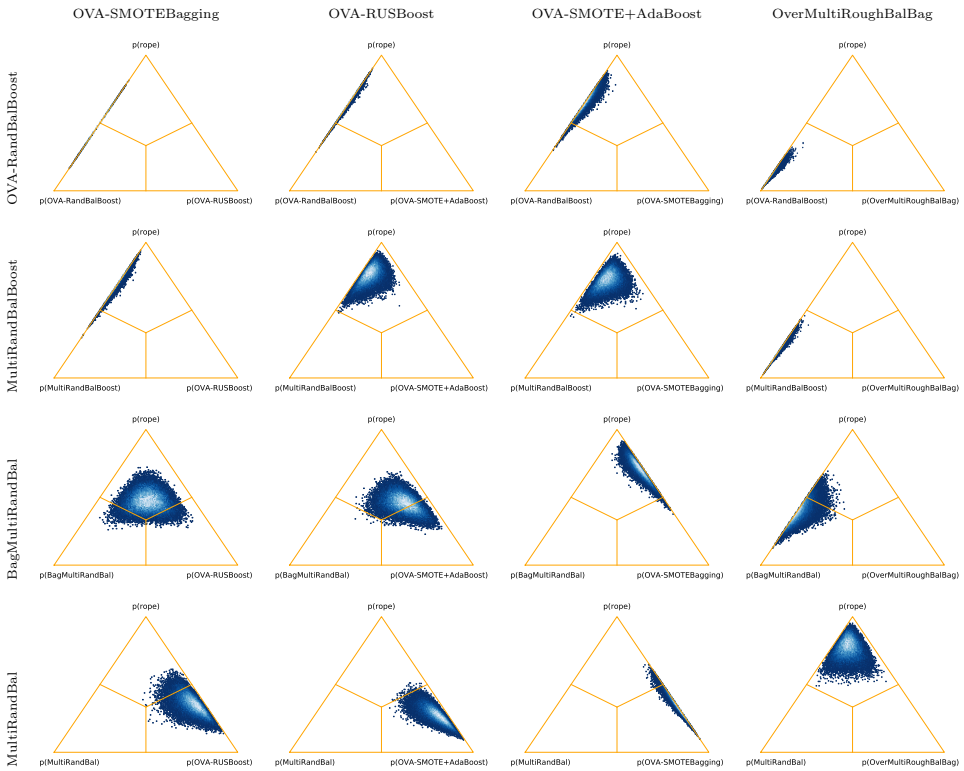


Figure 5: Posteriors for the Bayesian sign-rank tests, from Accuracy.

the diversity of a classifier with binarization as the average of the diversities of the ensembles for the binarized problems. Hence, diversities of multiclass ensembles and binarized ensembles may may not be commensurable.

Table 13 shows the average ranks for diversity. In general, Boosting-methods are more diverse than Bagging-based methods. This is consistent with the usual behaviour of Boosting and Bagging. For Random Balance, it can be observed that the three methods with RandBalBoost have more diversity than the three methods with BagRandBal, while these are more diverse than the three methods with RandBal.

Figure 11 shows diversity-performance diagrams, for some of the performance measures and the selected methods. The number of points in the scatter plots is the number of data sets. For this measure of diversity, smaller values indicate greater diversity. Hence, more diverse classifiers are at the left. It can be seen that methods based on boosting have more diverse classifiers.

5.4. Computation time

Table 14 summarises running times. The values of mean times across all the data sets depend heavily on a few data sets with much higher times, so the median times and average ranks are also shown. Methods that use Random Balance with binarization techniques are among the slowest, but they are comparable to other methods that use SMOTE such as SMOTEBagging. For Random Balance without binarization

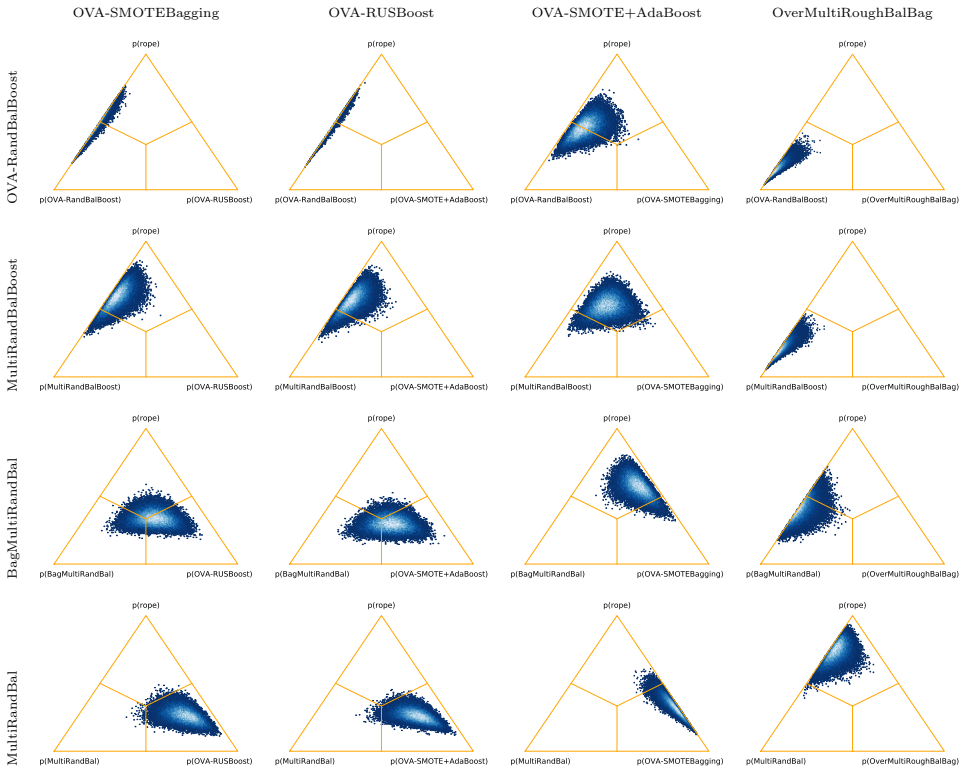


Figure 6: Posteriors for the Bayesian sign-rank tests, from Kappa.

Table 13: Average ranks for diversity

All methods

Method	Rank	$p_{Hochberg}$
OVA-RandBalBoost	3.0962	
OVA-RUSBoost	3.8077	0.622035
OVA-SMOTE+AdaBoost	4.1346	0.622035
MultiRandBalBoost	4.7885	0.622035
OVO-RandBalBoost	7.2788	0.015029
OVO-SMOTE+AdaBoost	7.8846	0.004541
UnderMultiRoughBalBag	9.2308	0.000128
OVO-BagRandBal	10.0962	0.000009
OVA-BagRandBal	10.1731	0.000008
BagMultiRandBal	11.2788	0.000000
OVO-RUSBoost	11.9231	0.000000
AdaBoost.NC	12.2885	0.000000
OVA-RoughBalBag	13.1635	0.000000
OVO-RoughBalBag	14.0385	0.000000
OVA-AdaBoost.NC	14.7115	0.000000
OVA-RandBal	15.6923	0.000000
MultiRandBal	16.2692	0.000000
OVO-RandBal	16.4808	0.000000
OVO-AdaBoost.NC	17.6058	0.000000
OVO-EasyEnsemble	18.0769	0.000000
OverMultiRoughBalBag	18.3846	0.000000
OVA-EasyEnsemble	18.8077	0.000000
OVO-SMOTEBagging	21.3846	0.000000
SMOTEBagging	22.0577	0.000000
OVA-SMOTEBagging	22.3462	0.000000

Bagging-based methods

Method	Rank	$p_{Hochberg}$
UnderMultiRoughBalBag	2.8462	
OVA-BagRandBal	3.0385	0.746033
OVO-BagRandBal	3.1154	0.746033
BagMultiRandBal	3.5962	0.619646
OVA-RoughBalBag	4.6346	0.010380
OVO-RoughBalBag	5.0962	0.000755
OverMultiRoughBalBag	6.9038	0.000000
OVO-SMOTEBagging	8.1923	0.000000
OVA-SMOTEBagging	8.7885	0.000000
SMOTEBagging	8.7885	0.000000

Boosting-based methods

Method	Rank	$p_{Hochberg}$
OVA-RandBalBoost	2.5577	
OVA-RUSBoost	2.7500	0.785650
OVA-SMOTE+AdaBoost	3.6731	0.229411
MultiRandBalBoost	4.2500	0.050095
OVO-RandBalBoost	5.2308	0.000627
OVO-SMOTE+AdaBoost	5.8462	0.000017
OVO-RUSBoost	7.2885	0.000000
AdaBoost.NC	8.0192	0.000000
OVA-AdaBoost.NC	8.9615	0.000000
OVO-EasyEnsemble	9.5192	0.000000
OVA-EasyEnsemble	9.6923	0.000000
OVO-AdaBoost.NC	10.2115	0.000000

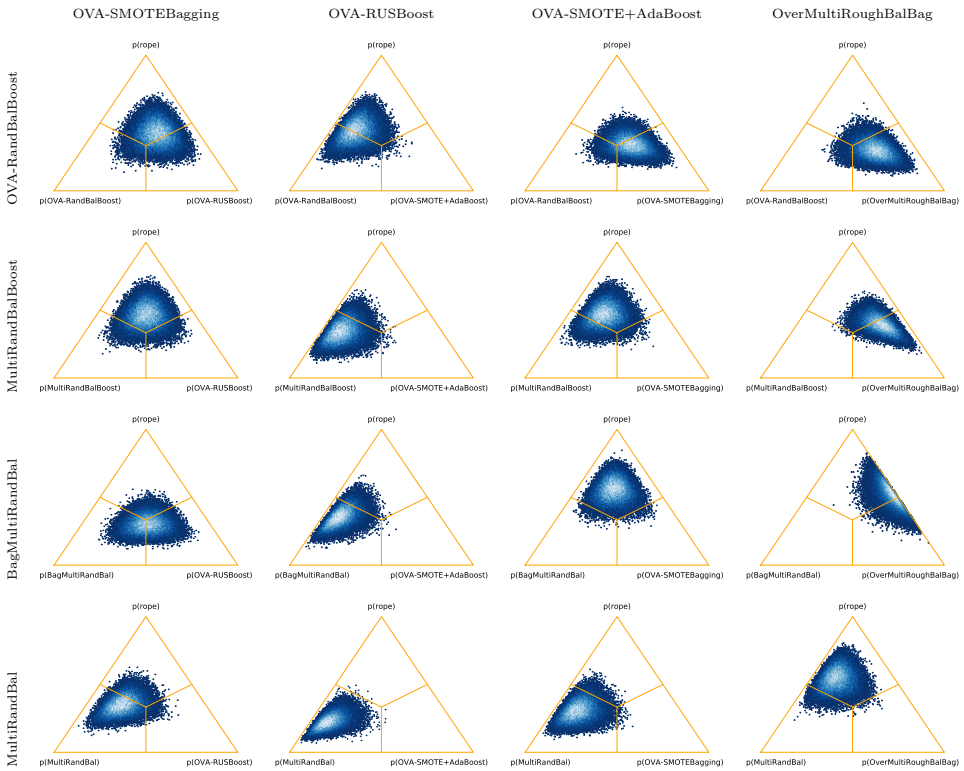


Figure 7: Posteriors for the Bayesian sign-rank tests, from G-mean.

techniques, times are more competitive.

6. Conclusion and Future Work

This study extends the Random Balance ensemble method to multiclass problems. We explored two extension routes. First, the original idea of abandoning the prior probabilities estimated from the class proportions, and sampling with randomly generated priors, is applied directly from 2 to c classes. Second, still using the random priors for two classes, we propose to decompose the c-class problem into a binary problem. The decomposition techniques which we adopted here are one-vs-one (OVO) and one-vs-all (OVA). Analysing six performance measures over a diverse collection of 52 data sets, we found that the configurations with Random Balance give better results than configurations that use state-of-the-art ensemble methods such as SMOTEBagging, RoughBalBag, SMOTE+AdaBoost, EasyEnsemble, RUSBoost and AdaBoost.NC with OVO and OVA.

The configurations with best results were OVA-RandBalBoost, MultiRandBalBoost, BagMultiRandBal and MultiRandBal. The last two have the advantage of being more efficient, as they are not based on building binary classifiers. Moreover, all the classifiers in the ensemble can be built in parallel, as the construction of one classifier does not depend on the results of others, as it happens with boosting. The use of OVA has been

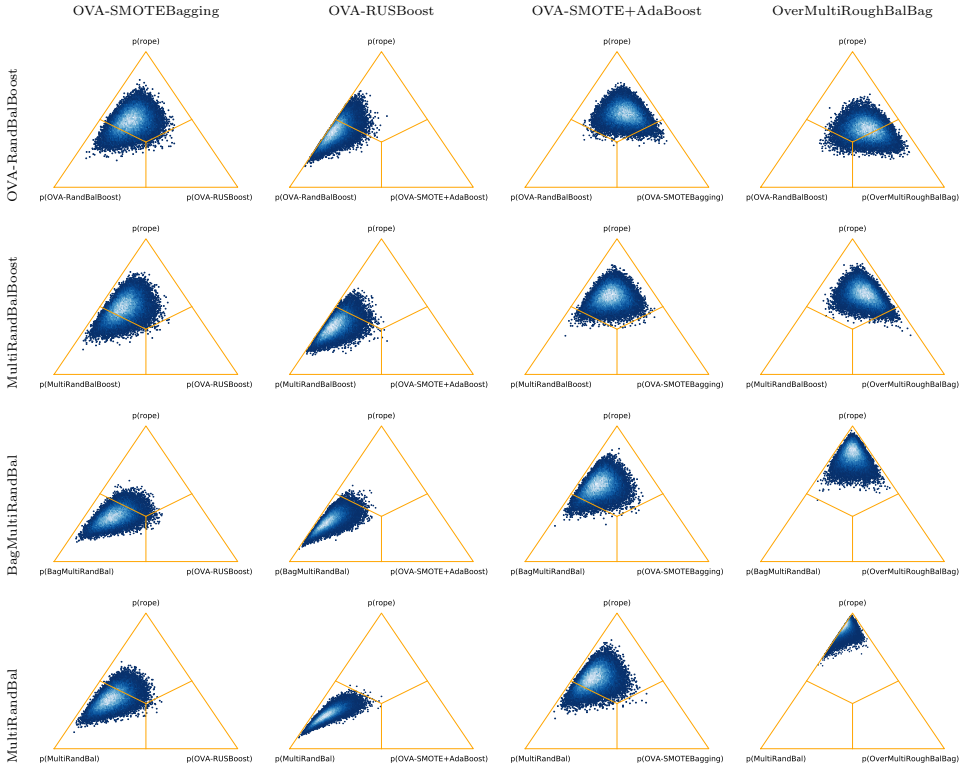


Figure 8: Posteriors for the Bayesian sign-rank tests, from average-Accuracy.

Table 14: Computation times

Training

Method	Rank	Mean	Median
UnderMultiRoughBalBag	1.0769	0.3228	0.0235
OverMultiRoughBalBag	3.0000	1.7727	0.1130
OVO-RoughBalBag	4.1731	2.0518	0.1485
OVA-RoughBalBag	4.5865	1.6500	0.1443
BagMultiRandBal	7.8558	12.1139	0.2723
OVA-EasyEnsemble	7.9519	1.9849	0.2492
AdaBoost.NC	8.1635	5.3272	0.3094
OVO-EasyEnsemble	8.1827	2.4865	0.3008
MultiRandBal	8.8077	12.2611	0.2954
OVO-AdaBoost.NC	8.9327	5.0472	0.3447
OVO-RUSBoost	9.5673	3.8526	0.3849
OVO-SMOTE+AdaBoost	10.5192	6.6301	0.3842
OVA-RUSBoost	11.8846	4.5546	0.4046
OVA-AdaBoost.NC	15.0962	14.3957	0.7015
OVA-SMOTE+AdaBoost	15.7500	18.3408	0.7124
MultiRandBalBoost	16.2500	54.0495	0.8855
SMOTEBagging	17.3654	92.0993	1.2384
OVO-BagRandBal	17.5673	95.4519	1.2389
OVO-RandBal	18.5962	94.2196	1.2863
OVO-SMOTEBagging	19.0962	109.2916	1.5082
OVO-RandBalBoost	19.7788	102.4023	1.3174
OVA-BagRandBal	21.3077	151.0985	1.6867
OVA-RandBal	21.9038	155.7964	1.8384
OVA-SMOTEBagging	23.4327	141.5879	2.6467
OVA-RandBalBoost	24.1538	185.4796	2.2729

Test

Method	Rank	Mean	Median
UnderMultiRoughBalBag	2.2885	0.0593	0.0061
BagMultiRandBal	3.6923	0.1173	0.0084
AdaBoost.NC	3.7500	0.1008	0.0074
MultiRandBal	3.8654	0.1083	0.0090
OverMultiRoughBalBag	5.0192	0.1164	0.0091
SMOTEBagging	8.8365	0.2127	0.0153
OVA-AdaBoost.NC	8.9423	0.5456	0.0135
OVO-AdaBoost.NC	10.4423	3.4941	0.0165
OVA-RoughBalBag	11.7596	0.4485	0.0177
OVA-BagRandBal	12.4038	0.4597	0.0173
MultiRandBalBoost	12.4904	0.2521	0.0185
OVA-RandBal	13.6058	0.4420	0.0188
OVA-SMOTE+AdaBoost	13.8654	0.7452	0.0220
OVA-EasyEnsemble	14.5192	0.5248	0.0214
OVO-SMOTE+AdaBoost	15.0481	4.6956	0.0243
OVA-SMOTEBagging	15.2308	0.4615	0.0205
OVO-RoughBalBag	16.4615	3.9608	0.0340
OVA-RUSBoost	17.4712	0.4961	0.0262
OVO-BagRandBal	17.5481	3.6663	0.0381
OVO-RandBal	17.7404	3.9682	0.0348
OVO-EasyEnsemble	17.8462	4.4325	0.0313
OVO-SMOTEBagging	18.3077	3.4955	0.0390
OVA-RandBalBoost	19.1538	0.7518	0.0271
OVO-RandBalBoost	22.2308	4.5419	0.0459
OVO-RUSBoost	22.4808	4.6437	0.0512

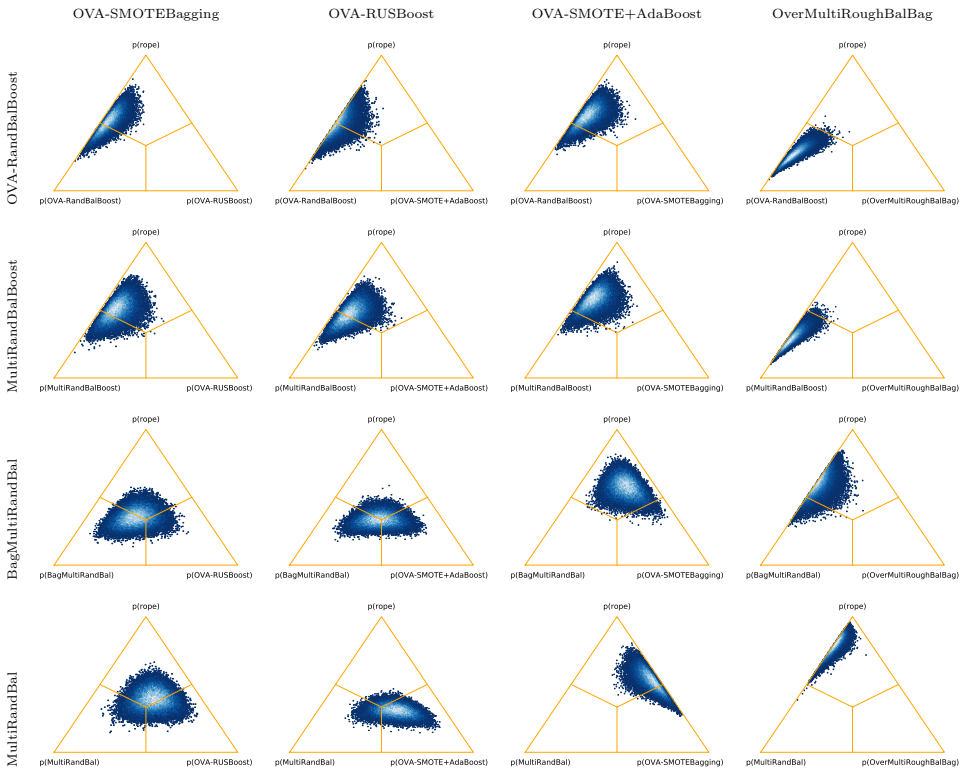


Figure 9: Posteriors for the Bayesian sign-rank tests, from F-measure.

previously considered not advisable for multiclass imbalance learning [25, 17] because in the decomposed binary problems, imbalance is artificially created (a balanced multiclass problem is solved through several imbalanced binary problems) or further increased. Our results, for both methods with and without Random Balance, contradict this advice. Table 9 suggests that using OVA in the context of Random Balance is more advantageous than using OVO, especially when the performance measure of choice is MAUC or if BaggingRandBal is adopted.

Random Balance ensembles are based on undersampling and oversampling. We chose here random undersampling and oversampling with SMOTE. More advanced approaches, such as ADASYN [52], evolutionary undersampling [71], cluster-based undersampling [26], ROSE [72], SMOTE-IPF [73], SMOM [30] could further improve the results of Random Balance.

We also chose the most widely-used decomposition techniques for multiclass problems: OVA and OVO. More advanced approaches could be considered. The classifiers obtained with OVA or OVO can be combined in different ways, such as DRCW-OVO [74], DRCW-ASEG [75] or the methods proposed in [76]. There are other approaches different from OVA and OVO, such as Error Correcting Output Codes [23, 46]. Dynamic ensemble selection [44] could also be applied with ensembles generated with Random Balance.

The results of ensemble methods depend on the method used to build the base classifiers. In this work,

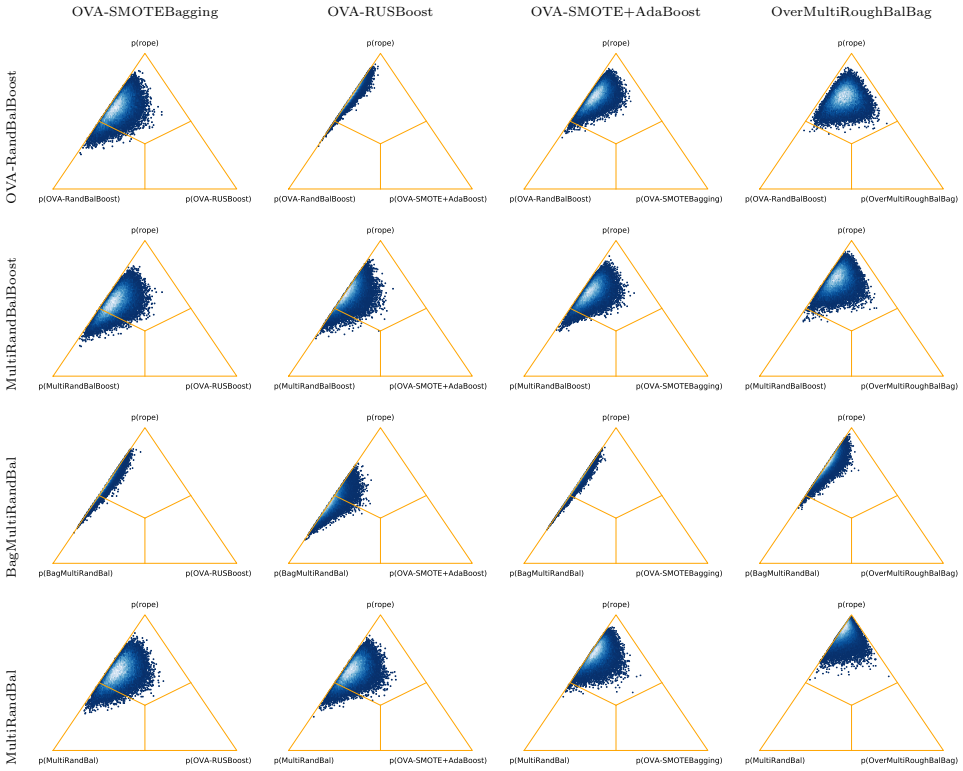


Figure 10: Posteriors for the Bayesian sign-rank tests, from MAUC.

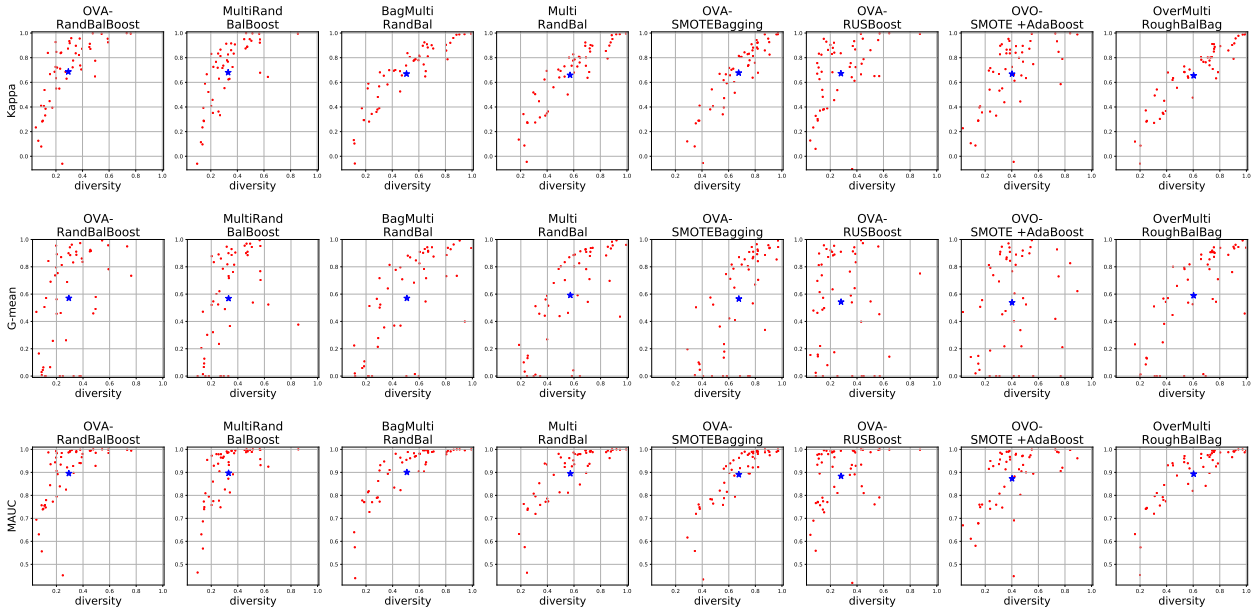


Figure 11: Diversity-performance diagrams. Each red dot is for a data set, the average value is marked with a blue star (⋆).

decision trees have been used, since they are very commonly used in ensembles. The proposed methods could be tested with other base classifiers. In particular, we could try classifiers with good results in recent comparisons [77], such as Extreme Learning Machine (ELM) or Sparse Representation based Classification (SRC). Moreover the proposed methods can be used with heterogeneous base classifiers as this approach has been reported to give good results [46].

The proposed RandBal extensions can be included in *Imbalanced-learn* [55] or *Multi-Imbalance* [56]. Also, they can be applied to ensemble methods that are not specifically designed for imbalance, for example, Stochastic Gradient Boosting Trees [78, 77].

Acknowledgements

This work was supported by the *Ministerio de Economía y Competitividad* of the Spanish Government through project TIN2015-67534-P (MINECO/FEDER, UE) and the *Junta de Castilla y León* through project BU085P17 (JCyL/FEDER, UE); both cofinanced from European Union FEDER funds.

References

[1] H. He, E. A. Garcia, Learning from imbalanced data, IEEE Transactions on Knowledge and Data Engineering 21 (9) (2009) 1263–1284. doi:10.1109/tkde.2008.239.

[2] P. Branco, L. Torgo, R. P. Ribeiro, A Survey of Predictive Modeling on Imbalanced Domains, ACM Comput. Surv. 49 (2). doi:10.1145/2907070.

[3] B. Krawczyk, Learning from imbalanced data: open challenges and future directions, Progress in Artificial Intelligence 5 (4) (2016) 221–232. doi:10.1007/s13748-016-0094-0.

[4] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, G. Bing, Learning from class-imbalanced data: Review of methods and applications, Expert Systems with Applications 73 (2017) 220 – 239. doi:10.1016/j.eswa.2016.12.035.
URL http://www.sciencedirect.com/science/article/pii/S0957417416307175

[5] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, F. Herrera, Learning from Imbalanced Data Sets, Springer, 2018. doi:10.1007/978-3-319-98074-4.

[6] Jie Sun and Hui Li and Hamido Fujita and Binbin Fu and Wenguo Ai, Class-imbalanced dynamic financial distress prediction based on Adaboost-SVM ensemble combined with SMOTE and time weighting, Information Fusion 54 (2020) 128 – 144. doi:https://doi.org/10.1016/j.inffus.2019.07.006.
URL http://www.sciencedirect.com/science/article/pii/S156625351830856X

[7] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, F. Herrera, Imbalanced Classification with Multiple Classes, Springer International Publishing, Cham, 2018, Ch. 8, pp. 197–226. doi:10.1007/978-3-319-98074-4_8.

[8] A. C. Tan, D. Gilbert, Y. Deville, Multi-class protein fold classification using a new ensemble machine learning approach, Genome informatics. International Conference on Genome Informatics 14 (2003) 206–217.
URL http://view.ncbi.nlm.nih.gov/pubmed/15706535

[9] X.-M. Zhao, X. Li, L. Chen, K. Aihara, Protein classification with imbalanced data, Proteins: Structure, Function, and Bioinformatics 70 (4) (2008) 1125–1132. doi:10.1002/prot.21870.

[10] T. W. Liao, Classification of weld flaws with imbalanced class data, Expert Systems with Applications 35 (3) (2008) 1041 – 1052. doi:10.1016/j.eswa.2007.08.044.
URL http://www.sciencedirect.com/science/article/pii/S0957417407003223

[11] P. Santos, J. Maudes, A. Bustillo, Identifying maximum imbalance in datasets for fault diagnosis of gearboxes, Journal of Intelligent Manufacturing 29 (2) (2018) 333–351. doi:10.1007/s10845-015-1110-0.

[12] N. Zarinabad, M. P. Wilson, S. K. Gill, K. A. Manias, N. P. Davies, A. C. Peet, Multiclass imbalance learning: Improving classification of pediatric brain tumors from magnetic resonance spectroscopy, Magn. Reson. Med. 77 (6) (2017) 2114–2124. doi:10.1002/mrm.26318.

[13] T. Sun, L. Jiao, J. Feng, F. Liu, X. Zhang, Imbalanced hyperspectral image classification based on maximum margin, IEEE Geoscience and Remote Sensing Letters 12 (3) (2015) 522–526. doi:10.1109/LGRS.2014.2349272.

[14] P. Pramokchon, P. Piamsa-nga, Reducing effects of class imbalance distribution in multi-class text categorization, in: Recent Advances in Information and Communication Technology, Springer, 2014, pp. 263–272.

[15] B. Fergani, et al., A new multi-class wsvm classification to imbalanced human activity dataset, Journal of Computers 9 (7) (2014) 1560–1565.

[16] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, F. Herrera, A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42 (4) (2012) 463–484. doi:10.1109/TSMCC.2011.2161285.

[17] Z. Zhang, B. Krawczyk, S. García, A. Rosales-Pérez, F. Herrera, Empowering one-vs-one decomposition with ensemble learning for multi-class imbalanced data, Knowledge-Based Systems 106 (2016) 251–263. doi:10.1016/j.knosys.2016.05.048.

[18] Z.-H. Zhou, X.-Y. Liu, On multi-class cost-sensitive learning, Computational Intelligence 26 (3) (2010) 232–257.

[19] L. I. Kuncheva, Combining Pattern Classifiers. Methods and Algorithms, 2nd Edition, John Wiley and Sons, 2014.

[20] J. F. Díez-Pastor, J. J. Rodríguez, C. García-Osorio, L. I. Kuncheva, Random balance: Ensembles of variable priors classifiers for imbalanced data, Knowledge-Based Systems 85 (2015) 96 – 111. doi:10.1016/j.knosys.2015.04.022.

[21] R. E. Schapire, The boosting approach to machine learning: An overview, in: Nonlinear estimation and classification, Springer, 2003, pp. 149–171.

[22] J. A. Sáez, B. Krawczyk, M. Woźniak, Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets, Pattern Recognition 57 (2016) 164–178. doi:10.1016/j.patcog.2016.03.012.

[23] O. Pujol, P. Radeva, J. Vitria, Discriminant ECOC: A heuristic method for application dependent design of error correcting output codes, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (6) (2006) 1007–1012.

[24] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes, Pattern Recognition 44 (8) (2011) 1761–1776. doi:10.1016/j.patcog.2011.01.017.

[25] W. Prachuabsupakij, N. Soonthornphisaj, Clustering and combined sampling approaches for multi-class imbalanced data classification, in: D. Zeng (Ed.), Advances in Information Technology and Industry Applications, Vol. 136 of Lecture Notes in Electrical Engineering, Springer Berlin Heidelberg, 2012, pp. 717–724. doi:10.1007/978-3-642-26001-8_91.

[26] A. Agrawal, H. L. Viktor, E. Paquet, SCUT: Multi-class imbalanced data classification using smote and cluster-based undersampling, in: 2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K), Vol. 01, 2015, pp. 226–234.

[27] L. Abdi, S. Hashemi, To combat multi-class imbalanced problems by means of over-sampling and boosting techniques, Soft Computing 19 (12) (2015) 3369–3385. doi:10.1007/s00500-014-1291-z.

[28] L. Abdi, S. Hashemi, To combat multi-class imbalanced problems by means of over-sampling techniques, IEEE Transactions on Knowledge and Data Engineering 28 (1) (2016) 238–251. doi:10.1109/TKDE.2015.2458858.

[29] X. Yang, Q. Kuang, W. Zhang, G. Zhang, Amdo: An over-sampling technique for multi-class imbalanced problems, IEEE Transactions on Knowledge and Data Engineering 30 (9) (2018) 1672–1685. doi:10.1109/TKDE.2017.2761347.

[30] T. Zhu, Y. Lin, Y. Liu, Synthetic minority oversampling technique for multiclass imbalance problems, Pattern Recognition 72 (2017) 327 – 340. doi:10.1016/j.patcog.2017.07.024.
URL http://www.sciencedirect.com/science/article/pii/S0031320317302947

[31] T. R. Hoens, Q. Qian, N. V. Chawla, Z.-H. Zhou, Building decision trees for the multi-class imbalance problem, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2012, pp. 122–134.

[32] M. Lin, K. Tang, X. Yao, Dynamic sampling approach to training neural networks for multiclass imbalance classification, IEEE Transactions on Neural Networks and Learning Systems 24 (4) (2013) 647–660. doi:10.1109/TNNLS.2012.2228231.

[33] D. Díaz-Vico, A. R. Figueiras-Vidal, J. R. Dorronsoro, Deep mlps for imbalanced classification, in: 2018 International Joint Conference on Neural Networks (IJCNN), IEEE, 2018, pp. 1–7.

[34] Y. Sun, M. S. Kamel, Y. Wang, Boosting for learning multiple classes with imbalanced class distribution, in: Sixth International Conference on Data Mining (ICDM'06), IEEE, 2006, pp. 592–602.

[35] B. Krawczyk, Cost-sensitive one-vs-one ensemble for multi-class imbalanced data, in: Neural Networks (IJCNN), 2016 International Joint Conference on, IEEE, 2016, pp. 2447–2452.

[36] Z.-L. Zhang, X.-G. Luo, S. García, F. Herrera, Cost-sensitive back-propagation neural networks with binarization techniques in addressing multi-class problems and non-competent classifiers, Applied Soft Computing 56 (2017) 357–367.

[37] S. Vluymans, A. Fernández, Y. Saeys, C. Cornelis, F. Herrera, Dynamic affinity-based classification of multi-class imbalanced data with one-versus-one decomposition: a fuzzy rough set approach, Knowledge and Information Systems 56 (1) (2018) 55–84.

[38] A. Fernández, V. López, M. Galar, M. J. del Jesus, F. Herrera, Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches, Knowledge-Based Systems 42 (2013) 97–110. doi:10.1016/j.knosys.2013.01.018.

[39] R. Barandela, R. Valdovinos, J. Sánchez, New applications of ensembles of classifiers, Pattern Analysis & Applications 6 (3) (2003) 245–256. doi:10.1007/s10044-003-0192-z.

[40] S. Wang, X. Yao, Diversity analysis on imbalanced data sets by using ensemble models, in: Computational Intelligence and Data Mining, 2009. CIDM'09. IEEE Symposium on, IEEE, 2009, pp. 324–331.

[41] C. Seiffert, T. Khoshgoftaar, J. Van Hulse, A. Napolitano, RUSBoost: A hybrid approach to alleviating class imbalance, Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on 40 (1) (2010) 185–197.

[42] N. Chawla, A. Lazarevic, L. Hall, K. Bowyer, SMOTEBoost: Improving prediction of the minority class in boosting, in: 7th European Conference on Principles and Practice of Knowledge Discovery in Databases(PKDD 2003), 2003, pp. 107–119.

[43] X. Y. Liu, J. Wu, Z. H. Zhou, Exploratory undersampling for class-imbalance learning, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 39 (2) (2009) 539–550. doi:10.1109/TSMCB.2008.2007853.

[44] S. García, Z.-L. Zhang, A. Altalhi, S. Alshomrani, F. Herrera, Dynamic ensemble selection for multi-class imbalanced datasets, Information Sciences 445-446 (2018) 22 – 37. doi:10.1016/j.ins.2018.03.002.

URL http://www.sciencedirect.com/science/article/pii/S0020025518301725
[45] A. Sen, M. M. Islam, K. Murase, X. Yao, Binarization with boosting and oversampling for multiclass classification, IEEE transactions on cybernetics 46 (5) (2016) 1078–1091.
[46] J. Bi, C. Zhang, An empirical comparison on state-of-the-art multi-class imbalance learning algorithms and a new diversified ensemble learning scheme, Knowledge-Based Systems 158 (2018) 81 – 93. doi:10.1016/j.knosys.2018.05.037.
URL http://www.sciencedirect.com/science/article/pii/S095070511830282X
[47] S. Chen, H. He, E. A. Garcia, RAMOBoost: ranked minority oversampling in boosting, IEEE Transactions on Neural Networks 21 (10) (2010) 1624–1642.
[48] S. Wang, X. Yao, Multiclass imbalance problems: Analysis and potential solutions, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 42 (4) (2012) 1119–1130. doi:10.1109/TSMCB.2012.2187280.
[49] G. Collell, D. Prelec, K. R. Patil, A simple plug-in bagging ensemble based on threshold-moving for classifying binary and multiclass imbalanced data, Neurocomputing 275 (2018) 330 – 340. doi:10.1016/j.neucom.2017.08.035.
URL http://www.sciencedirect.com/science/article/pii/S092523121731456X
[50] M. Lango, J. Stefanowski, Multi-class and feature selection extensions of roughly balanced bagging for imbalanced data, Journal of Intelligent Information Systems 50 (1) (2018) 97–127. doi:10.1007/s10844-017-0446-7.
[51] H. Han, W.-Y. Wang, B.-H. Mao, Borderline-smote: A new over-sampling method in imbalanced data sets learning, in: D.-S. Huang, X.-P. Zhang, G.-B. Huang (Eds.), Advances in Intelligent Computing: International Conference on Intelligent Computing, ICIC 2005, Hefei, China, August 23-26, 2005, Proceedings, Part I, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 878–887. doi:10.1007/11538059_91.
[52] H. He, Y. Bai, E. A. Garcia, S. Li, ADASYN: Adaptive synthetic sampling approach for imbalanced learning, in: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), 2008, pp. 1322–1328. doi:10.1109/IJCNN.2008.4633969.
[53] Y. Liu, X. Yao, Simultaneous training of negatively correlated neural networks in an ensemble, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 29 (6) (1999) 716–725. doi:10.1109/3477.809027.
[54] S. Hido, H. Kashima, Y. Takahashi, Roughly balanced bagging for imbalanced data, Statistical Analysis and Data Mining: The ASA Data Science Journal 2 (5-6) (2009) 412–426.
[55] G. Lemaître, F. Nogueira, C. K. Aridas, Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning, Journal of Machine Learning Research 18 (17) (2017) 1–5.
URL http://jmlr.org/papers/v18/16-365.html
[56] C. Zhang, J. Bi, S. Xu, E. Ramentol, G. Fan, B. Qiao, H. Fujita, Multi-imbalance: An open-source software for multi-class imbalance learning, Knowledge-Based Systems 174 (2019) 137 – 143. doi:10.1016/j.knosys.2019.03.001.
URL http://www.sciencedirect.com/science/article/pii/S0950705119301042
[57] L. Breiman, Bagging predictors, Machine Learning 26 (2) (1996) 123–140.
[58] N. Chawla, K. Bowyer, L. Hall, W. Kegelmeyer, Smote: Synthetic minority over-sampling technique, Journal of Artificial Intelligence Research 16 (2002) 321–357.
[59] J. Alcala-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera, Keel data-mining software tool: Data set repository and integration of algorithms and experimental analysis framework, Journal of Multiple-Valued Logic and Soft Computing 17 (2-3) (2011) 255–287.
[60] M. Fernández-Delgado, E. Cernadas, S. Barro, D. Amorim, Do we need hundreds of classifiers to solve real world classification problems?, Journal of Machine Learning Research 15 (2014) 3133–3181.
[61] D. Dheeru, E. Karra Taniskidou, UCI machine learning repository, http://archive.ics.uci.edu/ml (2017).
URL http://archive.ics.uci.edu/ml
[62] D. J. Hand, R. J. Till, A simple generalisation of the area under the roc curve for multiple class classification problems, Machine learning 45 (2) (2001) 171–186.
[63] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, The weka data mining software: an update, SIGKDD Explor. Newsl. 11 (1) (2009) 10–18. doi:10.1145/1656274.1656278.
[64] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.
[65] S. García, F. Herrera, An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons, Journal of Machine Learning Research 9 (2008) 2677–2694.
URL http://www.jmlr.org/papers/volume9/garcia08a/garcia08a.pdf
[66] Y. Hochberg, A sharper Bonferroni procedure for multiple tests of significance, Biometrika 75 (4) (1988) 800–802. doi:10.1093/biomet/75.4.800.
[67] A. Benavoli, G. Corani, J. Demšar, M. Zaffalon, Time for a change: a tutorial for comparing multiple classifiers through bayesian analysis, Journal of Machine Learning Research 18 (77) (2017) 1–36.
URL http://jmlr.org/papers/v18/16-305.html
[68] J. Quinlan, C4.5: Programs for Machine Learning, Morgan Kauffman, 1993.
[69] Y. Freund, R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences 55 (1) (1997) 119–139.
[70] C. Nadeau, Y. Bengio, Inference for the generalization error, Machine Learning 52 (3) (2003) 239–281.
[71] M. Galar, A. Fernández, E. Barrenechea, F. Herrera, EUSBoost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling, Pattern Recognition 46 (12) (2013) 3460–3471. doi:10.1016/j.patcog.2013.05.006.
[72] G. Menardi, N. Torelli, Training and assessing classification rules with imbalanced data, Data Mining and Knowledge Discovery 28 (1) (2014) 92–122. doi:10.1007/s10618-012-0295-5.
[73] J. A. Sáez, J. Luengo, J. Stefanowski, F. Herrera, SMOTE-IPF: addressing the noisy and borderline examples problem

in imbalanced classification by a re-sampling method with filtering, Information Sciences 291 (2015) 184 – 203. doi:10.1016/j.ins.2014.08.051.
URL http://www.sciencedirect.com/science/article/pii/S0020025514008561

[74] M. Galar, A. Fernández, E. Barrenechea, F. Herrera, DRCW-OVO: Distance-based relative competence weighting combination for one-vs-one strategy in multi-class problems, Pattern Recognition 48 (1) (2015) 28 – 42. doi:10.1016/j.patcog.2014.07.023.
URL http://www.sciencedirect.com/science/article/pii/S0031320314002829

[75] Z.-L. Zhang, X.-G. Luo, S. González, S. García, F. Herrera, Drcw-aseg: One-versus-one distance-based relative competence weighting with adaptive synthetic example generation for multi-class imbalanced datasets, Neurocomputing 285 (2018) 176 – 187. doi:10.1016/j.neucom.2018.01.039.
URL http://www.sciencedirect.com/science/article/pii/S0925231218300584

[76] L. Zhou, H. Fujita, Posterior probability based ensemble strategy using optimizing decision directed acyclic graph for multi-class classification, Information Sciences 400-401 (2017) 142 – 156. doi:10.1016/j.ins.2017.02.059.
URL http://www.sciencedirect.com/science/article/pii/S0020025516314207

[77] C. Zhang, C. Liu, X. Zhang, G. Almpanidis, An up-to-date comparison of state-of-the-art classification algorithms, Expert Systems with Applications 82 (2017) 128 – 150. doi:10.1016/j.eswa.2017.04.003.
URL http://www.sciencedirect.com/science/article/pii/S0957417417302397

[78] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, ACM, New York, NY, USA, 2016, pp. 785–794. doi:10.1145/2939672.2939785.